

Slepian functions on the sphere, generalized Gaussian quadrature rule

L Miranian

Department of Mathematics, University of California, Berkeley CA, 94720, USA

E-mail: luiza@math.berkeley.edu

Received 20 October 2003, in final form 1 March 2004

Published 2 April 2004

Online at stacks.iop.org/IP/20/877 (DOI: 10.1088/0266-5611/20/3/014)

Abstract

Denote by \mathbf{K} the operator of ‘time–band–time’ limiting on the surface of the sphere and consider the problem of computing singular vectors of \mathbf{K} . This problem can be reduced to a simpler task of computing eigenfunctions of a differential operator, if a differential operator, which commutes with \mathbf{K} and has a simple spectrum, can be exhibited. In Grünbaum *et al* (1982 *SIAM J. Appl. Math.* **42** 941–55) such a second-order differential operator commuting with \mathbf{K} on the appropriate subspaces was constructed. In this paper, this algebraic property of commutativity is used to produce an efficient numerical scheme for computing a convenient basis for the space of singular vectors of \mathbf{K} . The basis forms an extended Chebyshev system, and a generalized Gaussian quadrature rule for such a basis is presented.

1. Introduction

The fundamental problem of recovering a time-limited function from the knowledge of its Fourier transform on a certain band of frequencies is a central chapter in signal processing. This problem plays an important role in many aspects of image processing since it underlies the question of how to make optimal use of the available information that is always limited and corrupted by noise. The remarkable series of papers [2–8], by Slepian, Landau and Pollak in connection with the issue of time–band-limited signals has had a tremendous influence on many areas of engineering, science and mathematics. This work puts some of the pioneering work of Shannon [9] on firmer ground. Their starting points were fairly applied aspects of communication theory, optics, lasers, etc, but it became apparent that the ideas were applicable to many other situations.

The work presented in this paper deals with the case when the real line is replaced by the surface of the sphere. Here, the mathematical and computational issue is to get good approximations to the appropriate ‘Slepian functions’. In this instance ‘Slepian functions’ refer to a basis for the space of eigenfunctions of the operator \mathbf{K} which is obtained by the

successive application of the operations of time, band and time limiting. In the classical case of the real line, the computation of Slepian functions is done using the fact that the classical differential operator, resulting from the Laplacian by separation of variables in prolate spheroidal coordinates, happens to commute with the time–band-limiting integral operator. This commuting differential operator has a simple spectrum, hence its eigenfunctions form a basis for the space of eigenfunctions of the integral operator.

In the case of the surface of the sphere, it was shown in [1], that for a polar cap as well as for two symmetrically placed caps (one at each pole), a certain second-order differential operator

$$\mathbf{D} = \frac{d}{dx} \left[(1-x^2)(b-x) \frac{d}{dx} \right] - L(L+2)x - \frac{m^2(b-x)}{1-x^2}$$

defined on the interval $[b, 1]$ commutes with \mathbf{K} on the spaces of functions whose dependence on ϕ is of the form $e^{im\phi}$. The operator \mathbf{D} has a simple spectrum, hence its eigenfunctions, which are also eigenfunctions of \mathbf{K} , can serve as a basis for the space of eigenfunctions of \mathbf{K} . The same applies to the complement, in the sphere, of one or two polar caps. If the region in question has less symmetry, then one can always consider the integral operator, but the search for a commuting local operator has proved elusive.

This algebraic property of existence of a commutative differential operator holds the key to a good algorithm. Since the object to be produced is a basis for the time–band–time-limited functions on a certain region of the sphere, one needs to expand easily in this basis, which requires an efficient quadrature rule for evaluating inner products. It is well known that for a system of functions that forms an extended Chebyshev system, a generalized Gaussian quadrature rule always exists. In [10] a method of obtaining such a quadrature rule using the appropriate continuation scheme and well-chosen starting points for Newton’s method is described. The operator \mathbf{K} happens to be a finite rank Fredholm operator, which implies that it has only a finite number of non-zero eigenvalues. There are many ways to compute the null space of the operator \mathbf{K} , but the algebraic property discussed above makes it possible to replace the computation of the eigenfunctions of \mathbf{K} by the computation of the eigenfunctions of \mathbf{D} . This not only simplifies the task from the numerical point of view, but also produces orthogonal functions which form an extended Chebyshev system, assuring the existence of an efficient quadrature rule.

In order to numerically compute the eigenfunctions (of an appropriate self-adjoint extension) of \mathbf{D} we expand them in the basis of the shifted Legendre polynomials, and reduce the problem to the computation of generalized eigenvalues and eigenvectors of certain sparse matrices. We then use the method that has been advocated recently in [10]. The object produced is a good basis for the space of the eigenfunctions of \mathbf{K} in the case when the regions of interest are either a polar cap or a spherical belt bounded by two parallels.

Among the applications envisaged is the problem discussed in [11] involving gravity field missions. Due to launch conditions or engineering reasons, the sampling is done not on the whole surface of the earth but on one of the type of regions mentioned above. The computation of the so-called Slepian functions, in this case, has been attempted in the geodesy community without taking recourse to the mathematical and computational advantages that are derived from exploiting the results in [1] and the very recent note [12]. For other geodesy applications, see [13, 14]. In the first of these papers the region where the function is known is the surface of the oceans.

This paper is organized as follows. In section 2.1 the problem of computing eigenfunctions of the integral operator \mathbf{K} is discussed. In section 2.2 some properties of shifted Legendre polynomials are recalled. Sections 2.3 and 2.4 describe a method for computing Slepian

functions using the differential operator. In section 2.5 the generalized Gaussian quadrature rule for the Slepian functions on the spherical cap is presented and concluding remarks are in section 3.

2. Numerical computation of the eigenproblem

2.1. Direct computation of eigenfunctions of the integral operator \mathbf{K}

In this section, an attempt to compute eigenvalues/eigenvectors of the integral operator directly is described, and a more efficient alternative is suggested.

As discussed in [1], denote by A the ‘polar cap’ $0 \leq \theta \leq \arccos(b)$, $0 \leq \phi \leq 2\pi$. Then the operator \mathbf{K} is the ‘finite convolution integral operator’. Denote

$$u = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta),$$

then

$$(\mathbf{K}f)(u) = \int_A \sum_{l=0}^L P_l(\langle u, u' \rangle) f(u') du' = \int_A \left(\sum_{l=0}^L \sum_{m=-l}^l Y_{lm}(u) \overline{Y_{lm}(u')} \right) f(u') du',$$

where $Y_{lm}(u)$ is the usual ‘spherical harmonic’ with

$$\int_0^{2\pi} \int_{-1}^1 Y_{lm}(x, \phi) \overline{Y_{lm}(x, \phi)} dx d\phi = 1,$$

and

$$Y_{lm}(u) = Y_{lm}(\cos \theta, \phi) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(\cos \theta) e^{im\phi},$$

or

$$Y_{lm}(x, \phi) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(x) e^{im\phi}, \quad x = \cos \theta.$$

In the formulae above $P_l^m(x)$ denotes the associated Legendre polynomial.

The operator \mathbf{K} is a singular Fredholm operator with rank $(L+1)^2$, hence it has only $(L+1)^2$ non-zero eigenvalues. In the following proposition, some properties of the eigenvalues and eigenvectors of \mathbf{K} are summarized.

Proposition 1. *Consider the finite $(L+1)^2$ rank symmetric Fredholm operator as defined above and let H_m be the subspaces of functions on the polar cap whose ϕ dependence is given by $e^{im\phi}$. Then*

- (i) *There are $L+1$ linearly independent orthogonal eigenfunctions of \mathbf{K} that belong to the space H_0 ; consequently \mathbf{K} has only $L+1$ distinct non-zero eigenvalues that correspond to eigenfunctions in H_0 .*
- (ii) *The $L(L+1)/2$ non-zero eigenvalues of \mathbf{K} have multiplicity 2: in both subspaces H_m and H_{-m} \mathbf{K} has a simple spectrum, i.e. $L-m+1$ non-zero distinct eigenvalues, where $m = 1, \dots, L-1$.*
- (iii) *$L-|m|+1$ eigenfunctions of \mathbf{K} belong to H_m for all $|m| = 1, \dots, L-1$ and are orthogonal.*

Proof. Let us fix L and \bar{m} , and see what the integral operator looks like on $H_{\bar{m}}$, i.e. take $g(u) = f(\cos \theta) e^{i\bar{m}\phi} = f(x) e^{i\bar{m}\phi}$ and apply the operator \mathbf{K} to it:

$$\begin{aligned} \mathbf{K}f(u) &= \int_A \left(\sum_{l=0}^L \sum_{m=-l}^l Y_{lm}(u) \overline{Y_{lm}(u')} \right) g(u') du' \\ &= \int_0^{2\pi} \int_b^1 \left(\sum_{l=0}^L \sum_{m=-l}^l Y_{lm}(x, \phi) \overline{Y_{lm}(x', \phi')} \right) f(x') e^{i\bar{m}\phi'} dx' d\phi' \\ &= \int_0^{2\pi} \int_b^1 \left(\sum_{l=0}^L \sum_{m=-l}^l \frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!} P_l^m(x) e^{im\phi} P_l^m(x') e^{-im\phi'} \right) f(x') e^{i\bar{m}\phi'} dx' d\phi' \\ &= \int_b^1 \sum_{l=0}^L \sum_{m=-l}^l \frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!} P_l^m(x) P_l^m(x') f(x') \left(e^{im\phi} \int_0^{2\pi} e^{i\phi'(-m+\bar{m})} d\phi' \right) dx' \\ &= e^{i\bar{m}\phi} \int_b^1 \sum_{l=|\bar{m}|}^L \frac{2l+1}{2\pi} \frac{(l-|\bar{m}|)!}{(l+|\bar{m}|)!} P_l^{\bar{m}}(x) P_l^{\bar{m}}(x') f(x') dx'. \end{aligned}$$

To simplify the notation denote

$$K(m, x, x') = \sum_{l=|m|}^L \frac{2l+1}{2\pi} \frac{(l-|m|)!}{(l+|m|)!} P_l^m(x) P_l^m(x')$$

as the kernel of \mathbf{K} in subspace H_m .

The following observations can be made:

- (i) The associated Legendre polynomials P_l^m are linearly independent and $K(m, x, x') = K(m, x', x)$ defines a singular symmetric Fredholm operator which has only $L - |m| + 1$ distinct non-zero eigenvalues for all $|m| = 0, \dots, L - 1$.
- (ii) If $m = 0$, then \mathbf{K} has only $L + 1$ distinct non-zero eigenvalues in H_0 , and because of the symmetry of the kernel $K(0, x, x')$ the corresponding eigenvectors are orthogonal.
- (iii) $K(m, x, x') = K(-m, x, x')$. The symmetry of the kernel $K(m, x, x')$ implies that the eigenvectors of \mathbf{K} in H_m corresponding to different eigenvalues are orthogonal.
- (iv) If $m \neq 0$, then in both H_m and H_{-m} the operator \mathbf{K} has the same kernel, hence $L - |m| + 1$ eigenvalues of \mathbf{K} will be duplicated for every $|m| = 1, \dots, L - 1$. \square

Attempts to compute the eigenfunctions of the integral operator $\mathbf{K}F_n(u) = \mu_n F_n(u)$ directly have not been fruitful. In the experiments discussed below the integral operator with $L = 3$ was discretized using the Gaussian quadrature rule with $N_x = 21$ -point grid in $x = \cos(\theta)$ variable and $N_\phi = 21$ -point grid in ϕ variable. Let T be the matrix obtained after discretization of the integral operator \mathbf{K} . The disadvantages of this method are:

- (i) The size of the matrix T is $N_x N_\phi = 441$; it depends quadratically on the grid size, which makes computing its eigenvalues/eigenvectors an intensive task. In the alternative approach a grid of 10^4 points in the x variable is used, whereas in the direct approach it is computationally infeasible because of the reason just mentioned.
- (ii) Only eigenfunctions corresponding to non-zero eigenvalues could be computed directly, hence a procedure that would produce an orthogonal basis for the null space of \mathbf{K} is needed.

Below is the summary of some numerical experiments. Only the eigenfunctions of T that correspond to non-zero eigenvalues are meaningful, so only these are considered in the text below.

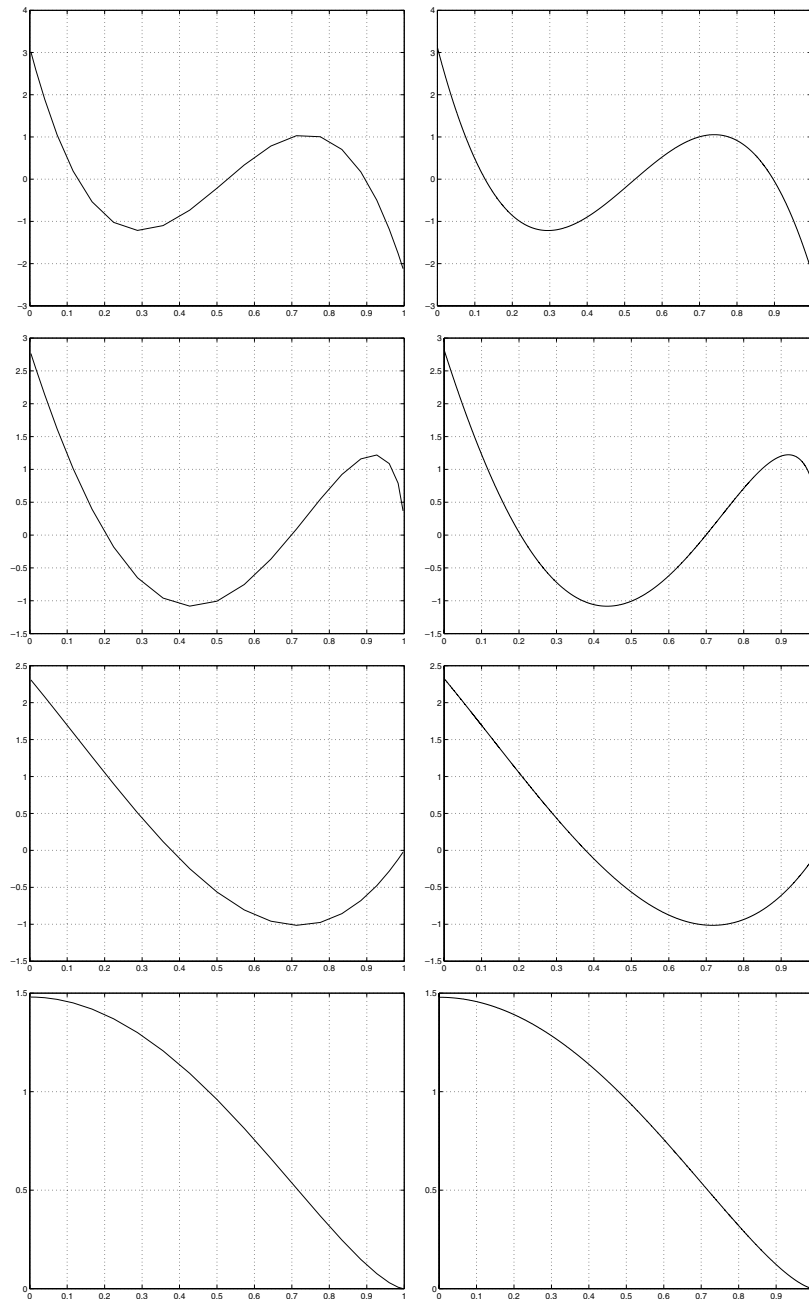


Figure 1. Results of the direct discretization (four left-hand figures) and eigenfunctions obtained by using the differential operator \mathbf{D} (four right-hand figures) for $m = 0, 1, 2, 3$. Functions presented are numbered according to the number of roots, i.e. G_3, G_2, G_1, G_0 ; $L = 3$.

- (i) Exactly four eigenfunctions of T are in subspace H_0 ; three eigenfunctions in $H_{\pm 1}$; two eigenfunctions in $H_{\pm 2}$ and one eigenfunction in $H_{\pm 3}$.
- (ii) Eigenfunctions $F_j(x) e^{\pm im\phi}$ with $j = 0, \dots, L - |m| + 1$ of T in each $H_{\pm m}$ are very close to these produced by the alternative procedure below (see figure 1).

(iii) Eigenfunctions of T corresponding to the eigenvalue 0 cannot be computed by direct discretization.

There is an efficient alternative to the procedure described above. It produces an orthonormal basis for the space of eigenfunctions of \mathbf{K} . Consider the following second-order differential operator on the interval $[b, 1]$:

$$\mathbf{D} = \frac{d}{dx} \left[(1-x^2)(b-x) \frac{d}{dx} \right] - L(L+2)x - \frac{m^2(b-x)}{1-x^2}.$$

A look at [1] will show that this is the appropriate operator \mathbf{D} that commutes with the \mathbf{K} built there when acting on H_m . The operator \mathbf{D} defines a Sturm–Liouville problem, hence it has a simple spectrum and orthogonal eigenfunctions. Because it commutes with \mathbf{K} we can say that the eigenfunctions of \mathbf{D} are also eigenfunctions of \mathbf{K} when acting on H_m . Since we search for an orthonormal basis in the space of eigenfunctions of \mathbf{K} , the eigenfunctions of \mathbf{D} provide us with such a basis.

In order to compute numerically the eigenfunctions (of an appropriate self-adjoint extension) of \mathbf{D} we expand them in the basis of the shifted Legendre polynomials, and reduce the problem to (generalized) eigenproblem for some well-structured matrices.

2.2. Shifted Legendre polynomials

In this section, an overview of some facts about shifted Legendre polynomials is presented. These facts will be used to design a procedure for computing eigenvalues/eigenvectors of the differential operator \mathbf{D} .

Define shifted Legendre polynomials to be the solutions of the following second-order differential equation:

$$(b-x)(1-x)S_n'' + 2(x-b_1)S_n' - n(n+1)S_n = 0.$$

Denote $b_1 := (1+b)/2$, $b_2 := (1-b)/2$. The following properties of S_n will be useful later:

(i) recursion relation

$$S_n = b_1 S_n + \frac{b_2(n+1)}{2n+1} S_{n+1} + \frac{b_2 n}{2n+1} S_{n-1};$$

(ii) derivative

$$(1-x)(b-x)S_n' = b_2 \frac{n(n+1)}{2n+1} (S_{n+1} - S_{n-1});$$

(iii) normalized shifted Legendre polynomials

$$\bar{S}_k \equiv S_k \sqrt{(2k+1)/2b_2}.$$

2.3. Computation of the eigenproblem $\mathbf{D}F_n = \lambda_n F_n$: case $m = 0$

In this section, the problem of computing the eigenfunctions of the differential operator \mathbf{D} with $m = 0$ is reduced to the problem of computing eigenvectors of a certain symmetric tridiagonal matrix.

Consider the following eigenproblem:

$$\left(\frac{d}{dx} \left[(1-x^2)(b-x) \frac{d}{dx} \right] - L(L+2)x \right) F_n = \lambda_n F_n. \quad (1)$$

Let $(a_0^n, a_1^n, a_2^n, \dots)$ be the coefficients of the expansion of $F_n(x)$ in the basis of shifted Legendre polynomials

$$F_n = \sum_{k=0}^{\infty} a_k^n S_k.$$

After substituting $F_n(x)$ into (1), using properties (i), (ii) and the linear independence of S_k one obtains a recursion relation

$$C_{k-1}a_{k-1} + B_{k+1}a_{k+1} + A_k a_k - \lambda_n a_k = 0, \quad (2)$$

where

$$\begin{aligned} A_k &= k(k+1)(1+b_1) - L(L+2)b_1 \\ B_k &= \frac{k}{2k+1} b_2 [(k-1)(k+1) - L(L+2)] \\ C_k &= \frac{k+1}{2k+1} b_2 [k(k+2) - L(L+2)]. \end{aligned}$$

After rewriting (2) for the normalized polynomials using property (iii), the recursion relation can be written in the matrix form as

$$M\bar{a}^n = \lambda_n \bar{a}^n, \quad (3)$$

where $\bar{a}^n = (\bar{a}_0^n, \bar{a}_1^n, \dots)^T$ are the coefficients of the expansion in the basis of the normalized shifted Legendre polynomials;

$$\begin{aligned} M_{k,k} &= k(k+1)(1+b_1) - L(L+2)b_1, \\ M_{k,k+1} &= \frac{b_2(k+1)}{\sqrt{(2k+3)(2k+1)}} [k(k+2) - L(L+2)], \\ M_{k+1,k} &= \frac{b_2(k+1)}{\sqrt{(2k+3)(2k+1)}} [k(k+2) - L(L+2)], \end{aligned}$$

$k = 0, 1, \dots$ and the remainder of the entries of the matrix being zero. Using \bar{a}^n obtained from (3), we express $F_n = \sum_{k=0}^{\infty} \bar{a}_k^n \bar{S}_k$. The numerical evidence strongly suggests that the coefficients \bar{a}_k^n decay very fast. In practice, this allows us to compute $F_n = \sum_{k=0}^N \bar{a}_k^n \bar{S}_k$, for certain large values of N .

2.4. Computation of the eigenproblem $\mathbf{D}G_n = \mu_n G_n$: case $m > 0$

In this section, the problem of computing eigenfunctions of the differential operator \mathbf{D} with $m > 0$ is reduced to a generalized matrix eigenproblem.

Consider the following eigenproblem:

$$[(1-x^2)(b-x)G_n']' - L(L+2)xG_n - \frac{m^2(b-x)}{1-x^2}G_n = \mu_n G_n. \quad (4)$$

Introduce the following simplifying notation:

- (i) Matrix $S = (\bar{S}_0, \bar{S}_1, \bar{S}_3, \dots)$, where \bar{S}_k are normalized shifted Legendre polynomials.
- (ii) Matrix A with columns $A^k = (\bar{a}_0^k, \bar{a}_1^k, \bar{a}_2^k, \dots)^T$, where \bar{a}_j^k are the coefficients of the expansion of $F_k(x)$ in the basis of normalized shifted Legendre polynomials.
- (iii) $F_k = \sum_{j=0}^{\infty} \bar{S}_j \bar{a}_j^k = SA^k$.

(iv) Also,

$$\begin{aligned} xF_k &= \sum_{j=0}^{\infty} xS_j a_j^k = \sum_{j=0}^{\infty} a_j^k \left(b_1 S_j + \frac{b_2(j+1)}{2j+1} S_{j+1} + \frac{b_2 j}{2j+1} S_{j-1} \right) \\ &= \sum_{j=0}^{\infty} \bar{a}_j^k \left(b_1 \bar{S}_j + \frac{b_2(j+1)}{\sqrt{(2j+1)(2j+3)}} \bar{S}_{j+1} + \frac{b_2 j}{\sqrt{(2j+1)(2j-1)}} \bar{S}_{j-1} \right) \\ &= SQA^k, \end{aligned}$$

where the symmetric tridiagonal matrix Q has entries

$$\begin{aligned} Q_{k,k} &= b_1, \\ Q_{k,k+1} &= Q_{k+1,k} = \frac{b_2(k+1)}{\sqrt{(2k+3)(2k+1)}}, \end{aligned}$$

with $= 0, 1, 2, \dots$. Similarly $x^2 F_k = SQ^2 A^k$.

(v) Denote $c^n = (c_0^n, c_1^n, c_2^n, \dots)^T$.

(vi) Matrix $\Omega = \text{diag}(\lambda_0, \lambda_1, \dots)$; $A\Omega = MA$.

(vii) Denote I to be the identity matrix.

Since the functions F_k form an orthonormal basis for $L_2[b, 1]$, one could expand G_n in the basis of F_k , i.e.

$$G_n = \sum_{k=0}^{\infty} c_k^n F_k(x) = \sum_{k=0}^{\infty} c_k^n SA^k = SA c^n = \sum_{k=0}^{\infty} \gamma_k^n \bar{S}_k(x), \quad (5)$$

where $\gamma^n = (\gamma_0^n, \gamma_1^n, \dots)^T = A c^n$. After substituting (5) into (4) and using the fact that F_k satisfies (1) one obtains

$$\sum c_k^n \left(\lambda_k F_k - \frac{m^2(b-x)}{1-x^2} F_k - \mu_n F_k \right) = 0,$$

or

$$\sum_{k=0}^{\infty} c_k^n (\lambda_k F_k (1-x^2) - m^2(b-x) F_k - \mu_n F_k (1-x^2)) = 0. \quad (6)$$

Using the notation above, (6) can be written as

$$\begin{aligned} &\sum_{k=0}^{\infty} c_k^n (\lambda_k S(I - Q^2)A^k - m^2 S(b - Q)A^k - \mu_n S(I - Q^2)A^k) \\ &= S \sum_{k=0}^{\infty} c_k^n (\lambda_k (I - Q^2)A^k - m^2(b - Q)A^k - \mu_n (I - Q^2)A^k) = 0. \end{aligned}$$

Since the functions S_k are linearly independent, one obtains

$$\begin{aligned} 0 &= \sum_{k=0}^{\infty} c_k^n (\lambda_k (I - Q^2)A^k - m^2(b - Q)A^k - \mu_n (I - Q^2)A^k) \\ &= (I - Q^2)A\Omega c^n - m^2(b - Q)Ac^n - \mu_n (I - Q^2)Ac^n \\ &= (I - Q^2)MAc^n - m^2(b - Q)Ac^n - \mu_n (I - Q^2)Ac^n \\ &= ((I - Q^2)M - m^2(b - Q) - \mu_n (I - Q^2))Ac^n, \end{aligned}$$

which implies the generalized eigenproblem

$$(I - Q^2)(M - \mu_n I)\gamma^n = m^2(b - Q)\gamma^n. \tag{7}$$

Although the coefficients γ^n can be computed from the elegant matrix eigenproblem (7), numerical experiments have shown that the coefficients of the expansion $G_n(x) = \sum_{k=0}^\infty \gamma_k^n \bar{S}_k(x)$ decay slowly. However, the scheme above can be significantly improved, as we explain now. Taking into account boundary conditions, write $G_n = (1 - x^2)^{m/2} g_n(x)$, for a certain function $g_n(x)$. After rewriting (4) in terms of $g_n(x)$ we arrive at the following eigenproblem:

$$[(1 - x^2)(b - x)g_n']' + 2mx(x - b)g_n' + ((m^2 + 2m - L^2 - 2L)x - mb(m + 1))g_n = \mu_n g_n. \tag{8}$$

Now, the scheme described above can be applied to the function $g_n(x) = \sum_{k=0}^\infty c_k^n F_k$ for some coefficients c_k^n using the differential equation (8). In this case the derivation of the generalized matrix eigenproblem is similar to that performed at the beginning of this section. After elaborate calculations one arrives at

$$(I - Q)(M - \mu_n I)\alpha^n = (mb(m + 1)(I - Q) + 2mQ_3 - m(m + 2)(Q - Q^2))\alpha^n, \tag{9}$$

where

$$\begin{aligned} \tilde{Q}_3(k, k) &= \frac{-b_2}{(2k + 3)(2k - 1)}, \\ \tilde{Q}_3(k + 1, k) &= -\tilde{Q}_3(k, k + 1) = \frac{b_1}{(2k + 3)(2k - 1)}, \\ \tilde{Q}_3(k, k + 2) &= \frac{-b_2(k + 1)}{(2k + 3)\sqrt{(2k + 1)(2k + 5)}}, \\ \tilde{Q}_3(k + 2, k) &= \frac{b_2(k + 2)}{(2k + 3)\sqrt{(2k + 1)(2k + 5)}}, \\ E(k, k) &= b_2k(k + 1), \quad Q_3 = \tilde{Q}_3E \end{aligned}$$

for $k = 0, 1, 2, 3, \dots$; matrices Q and M were defined before, and $Ac^n = \alpha^n$. Observe that

$$g_n(x) = \sum_{k=0}^\infty c_k^n F_k(x) = \sum_{k=0}^\infty c_k^n SA^k = S \sum_{k=0}^\infty c_k^n A^k = SAc^n = S\alpha^n,$$

which means that while c^n are coefficients of the expansion of $g_n(x)$ in the basis of $F_k(x)$'s, α^n 's are coefficients of the expansion of $g_n(x)$ in the basis of normalized shifted Legendre polynomials S_k 's. From (9) we can compute $g_n = \sum_{k=0}^\infty \alpha_k^n \bar{S}_k$, and obtain $G_n(x) = (1 - x^2)^{m/2} g_n(x)$.

Experiments suggest that coefficients of the expansions of $g_n = \sum_{k=0}^\infty \alpha_k^n \bar{S}_k$ decay very rapidly. Moreover, it is not hard to see that if $G_n(x) = (1 - x^2)^{m/2} g_n(x) = (1 - x^2)^{m/2} \sum_{k=0}^\infty \alpha_k^n \bar{S}_k$ is an eigenfunction of \mathbf{K} that corresponds to a non-zero eigenvalue, then $\alpha_p^n = 0$ for all $p > L - m$. In order to observe this recall that on the subspace H_m kernel of \mathbf{K} is $K(x, x') = (1 - x^2)^{m/2}(1 - x'^2)^{m/2}Z_{L-m}(x, x')$, where $Z_{L-m}(x, x')$ is a symmetric polynomial in x and x' of degree $L - m$. Then,

$$\begin{aligned} |\alpha_p^n| &= \left| \int_b^1 \frac{\bar{S}_p(x)G_n(x)}{(1 - x^2)^{m/2}} dx \right| = \left| \int_b^1 \frac{\bar{S}_p(x)}{(1 - x^2)^{m/2}} \frac{1}{\xi_n} \left(\int_b^1 K(m, x, x')G_n(x') dx' \right) dx \right| \\ &= \frac{1}{|\xi_n|} \left| \int_b^1 G_n(x') \left(\int_b^1 \frac{\bar{S}_p(x)}{(1 - x^2)^{m/2}} K(m, x, x') dx \right) dx' \right| \end{aligned}$$

$$\begin{aligned} &\leq \frac{1}{|\xi_n|} \sqrt{\int_b^1 \left(\int_b^1 \frac{K(m, x, x')}{(1-x^2)^{m/2}} \bar{S}_p(x) dx \right)^2 dx'} \\ &= \frac{1}{|\xi_n|} \sqrt{\int_b^1 \left(\int_b^1 (1-x'^2)^{m/2} Z_{L-m}(x, x') \bar{S}_p(x) dx \right)^2 dx'} = 0 \end{aligned}$$

for all $p > L - m$, since the Legendre polynomial $\bar{S}_p(x)$ of degree p is orthogonal to x^k for all $k < p$.

Because of the very rapid decay of the coefficients α_k^n , eigenfunctions G_n can be computed efficiently. It is very important to sort eigenvalues (along with corresponding eigenvectors) of (9) in the ascending order before computing the sum $g_n = \sum_{k=0}^N \bar{a}_k^n \bar{S}_k$ for some appropriate finite N .

In figure 2 one can see the decay of the coefficients of the expansion in the case $m = 1, L = 1$ and $b = 0$ for the eigenfunctions $G_5, G_{10}, G_{30}, G_{80}$; $m = 4, L = 7$ and $b = -1/2$ for the eigenfunction $G_5, G_{10}, G_{30}, G_{80}$.

In figure 3 eigenfunctions G_k of \mathbf{D} are presented, for $k = 1, 5, 10, 20, 35, 50, L = 5, m = 2, b = 1/2$.

2.5. Construction of generalized Gaussian quadratures for the eigenfunctions of \mathbf{D}

In this section, an algorithm for constructing the generalized Gaussian quadrature rule for the eigenfunctions of \mathbf{D} is described and results of some numerical experiments are presented.

Functions $G_n(x)$ form a complete orthonormal basis for the space $\mathbf{L}_2[b, 1]$, and a quadrature rule for computing integrals of the form $\int_b^1 f(x) G_n(x) dx$ efficiently for various functions $f(x)$ is needed. In this instance, an 'efficient quadrature rule' refers to a quadrature such that the error decreases at least exponentially as a function of the number of nodes used in the integration.

The eigenfunctions G_n form an extended Chebyshev system, and according to the principal result of [16], there exists a unique n -point generalized Gaussian quadrature rule with weight $W : [b, 1] \rightarrow \mathbf{R}^+$. Nodes (x_1, \dots, x_n) and weights (w_1, \dots, w_n) of the quadrature satisfy a system of nonlinear equations

$$\sum_{i=1}^n w_i G_j(x_i) = \int_b^1 G_j(x) W(x) dx; \quad j = 0, \dots, 2n - 1. \quad (10)$$

Newton's method for this system of equations is always quadratically convergent, since the fact that eigenfunctions form an extended Chebyshev system implies the Jacobian of the system being always nonsingular ([15, lemma 2.6]).

In order to provide a good starting point for Newton's method we use the continuation scheme suggested in [15]. A numerical algorithm for obtaining weights and nodes of n -point generalized Gaussian quadrature is described below.

- (i) As a starting point of the algorithm take the n roots of G_n , denote them as $r = (r_1, \dots, r_n)$.
- (ii) Use a continuation scheme, i.e. let the weight functions $\omega: [0, 1] \times [b, 1] \rightarrow \mathbf{R}^+$ be defined by the formula

$$\omega(\alpha, x) = \alpha W(x) + (1 - \alpha) \sum_{j=1}^n \delta(x - r_j),$$

where δ denotes the Dirac delta function. Observe that when $\alpha = 1$, the weight function is equal to the desired weight function $W(x)$, and when $\alpha = 0$ the Gaussian weights and nodes are $w_i = 1, x_i = r_i$ for $i = 1, \dots, n$.

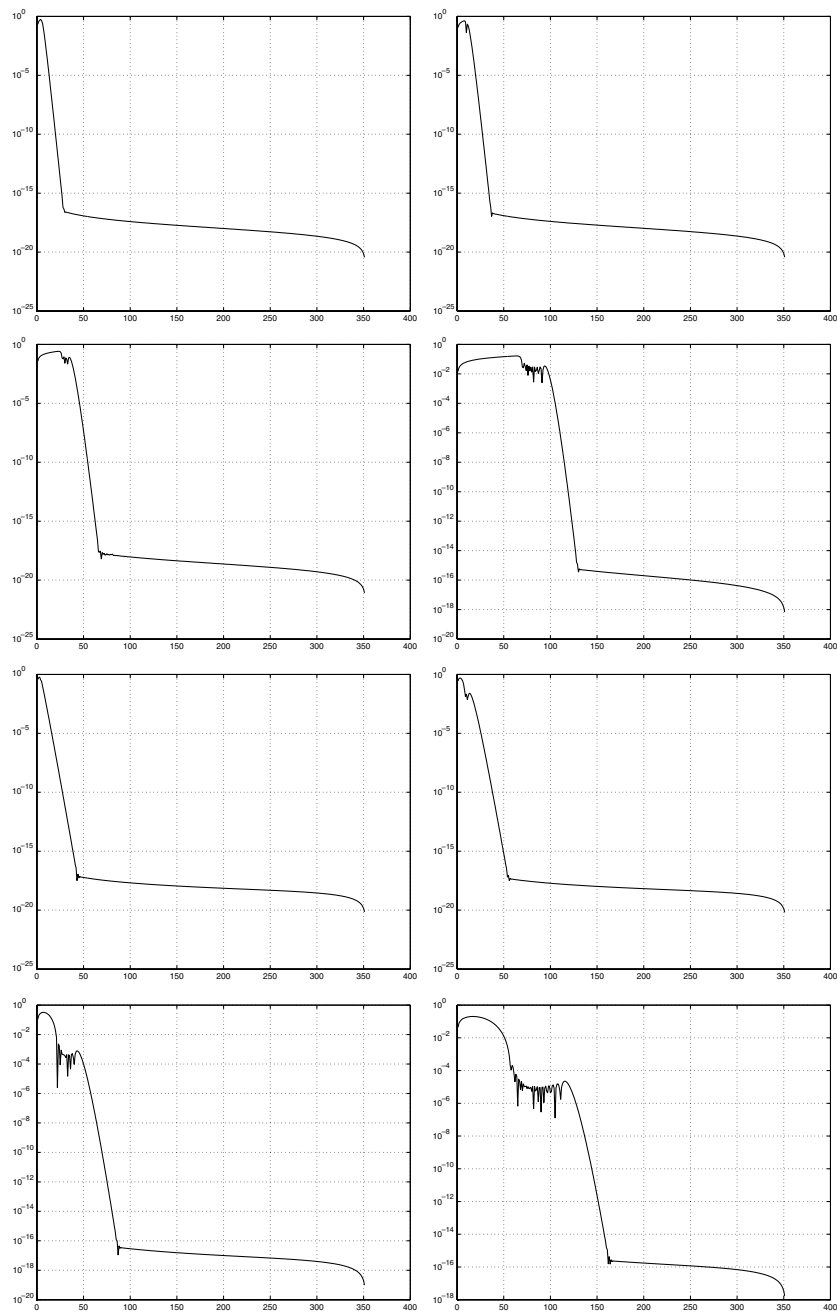


Figure 2. Top four plots: $L = 1, m = 1, b = 0$; bottom four plots: $L = 7, m = 4, b = -1/2$; graphs correspond to absolute values of the first 350 coefficients of the expansions for eigenfunctions $G_5, G_{10}, G_{30}, G_{80}$ versus their index.

- (iii) Damped Newton's method (at every iteration of Newton's method search for an appropriate step size along the direction prescribed by regular Newton's method) is used to solve the system on every step of the continuation scheme.

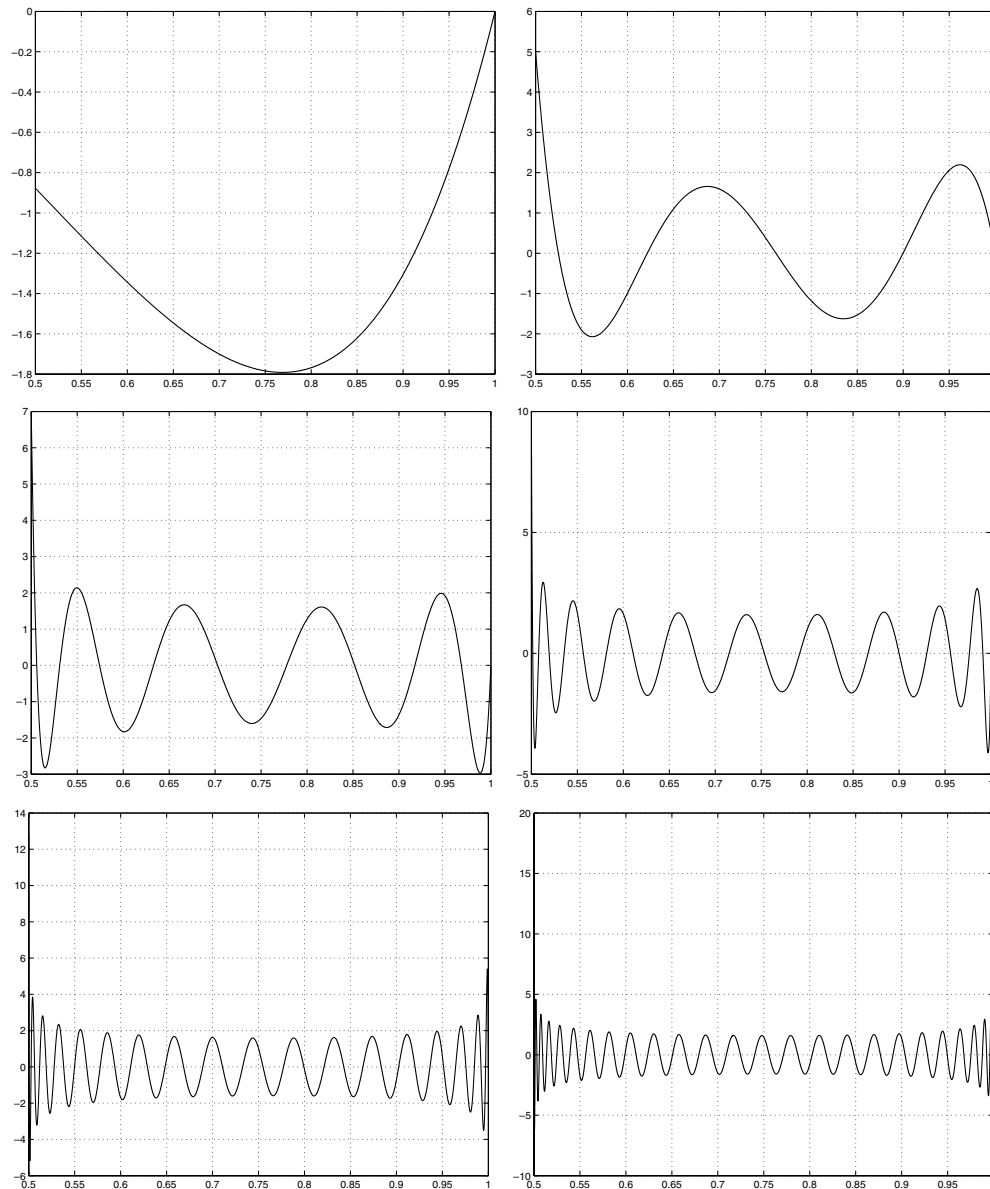


Figure 3. Eigenfunctions G_k of \mathbf{D} for $k = 1, 5, 10, 20, 35, 50$; $b = 1/2$, $L = 5$, $m = 2$.

In the numerical experiments conducted using MATLAB, the quadrature rule (10) for various weight functions was obtained. In particular, quadratures with $W(x) = 1$, $\sqrt{x-b}$ were constructed and the resulting weights and nodes are presented in tables 1 and 2.

To see the accuracy of the quadratures, the test function $f(x) = \sin^5(x) \cos(7x)(1-x^2)^2$, for instance, is integrated with relative and absolute errors of 7.08×10^{-10} and 2.40×10^{-16} with a 15-point quadrature, where $W(x) = 1$. The test function $f(x) = x^2(x-b)^{3/2}(1-x^2)^{5/2}$ is integrated with relative and absolute errors of 5.65×10^{-10} and 1.04×10^{-11} with a 10-point quadrature, where $W(x) = \sqrt{x-b}$.

Table 1. Nodes and weights of a $n = 5$ -, 10 -, 20 -point generalized Gaussian quadrature; $L = 4$; $m = 2$; $b = 0$; $W(x) = 1$.

N	Nodes x_i	Weights w_i
5	$3.061430650969785 \times 10^{-2}$	$7.837570652165048 \times 10^{-2}$
	$1.591475526561901 \times 10^{-1}$	$1.767635421198835 \times 10^{-1}$
	$3.761709472998213 \times 10^{-1}$	$2.506589661618823 \times 10^{-1}$
	$6.408517480864147 \times 10^{-1}$	$2.658094013325130 \times 10^{-1}$
	$8.780822138328971 \times 10^{-1}$	$1.932312576068380 \times 10^{-1}$
10	$8.604975547632166 \times 10^{-3}$	$2.207769315114833 \times 10^{-2}$
	$4.527940410382819 \times 10^{-2}$	$5.123752625510280 \times 10^{-2}$
	$1.108701381097156 \times 10^{-1}$	$7.975363584427851 \times 10^{-2}$
	$2.041512383172495 \times 10^{-1}$	$1.063292519876557 \times 10^{-1}$
	$3.221379242071772 \times 10^{-1}$	$1.286970636980550 \times 10^{-1}$
	$4.590563702382297 \times 10^{-1}$	$1.435765250777324 \times 10^{-1}$
	$6.055072094196928 \times 10^{-1}$	$1.471171930360809 \times 10^{-1}$
	$7.483505046653010 \times 10^{-1}$	$1.358982670708245 \times 10^{-1}$
	$8.718162722550230 \times 10^{-1}$	$1.083234916072349 \times 10^{-1}$
	$9.600009126676842 \times 10^{-1}$	$6.58775990056947 \times 10^{-2}$
20	$2.313389565053658 \times 10^{-3}$	$5.936750445069434 \times 10^{-3}$
	$1.218752444025983 \times 10^{-2}$	$1.381563354931116 \times 10^{-2}$
	$2.994246439820815 \times 10^{-2}$	$2.169029360224803 \times 10^{-2}$
	$5.555435092225135 \times 10^{-2}$	$2.952299561392022 \times 10^{-2}$
	$8.895761623636096 \times 10^{-2}$	$3.726213368537990 \times 10^{-2}$
	$1.300203295751197 \times 10^{-1}$	$4.482511676477427 \times 10^{-2}$
	$1.785092814999323 \times 10^{-1}$	$5.209093302886898 \times 10^{-2}$
	$2.340488118981268 \times 10^{-1}$	$5.889506175213475 \times 10^{-2}$
	$2.960759639570856 \times 10^{-1}$	$6.502739863025273 \times 10^{-2}$
	$3.637955855843940 \times 10^{-1}$	$7.023445222659594 \times 10^{-2}$
	$4.361403335253513 \times 10^{-1}$	$7.422717345129071 \times 10^{-2}$
	$5.117418243587686 \times 10^{-1}$	$7.669557687323547 \times 10^{-2}$
	$5.889200757075352 \times 10^{-1}$	$7.733072212747030 \times 10^{-2}$
	$6.656984698634406 \times 10^{-1}$	$7.585359103842103 \times 10^{-2}$
	$7.398503206783987 \times 10^{-1}$	$7.204896745102966 \times 10^{-2}$
	$8.089804217861422 \times 10^{-1}$	$6.580079947124912 \times 10^{-2}$
	$8.706406686725344 \times 10^{-1}$	$5.712405693982534 \times 10^{-2}$
	$9.224733629316755 \times 10^{-1}$	$4.618724004952186 \times 10^{-2}$
	$9.623698984922738 \times 10^{-1}$	$3.331967754066058 \times 10^{-2}$
	$9.886262004970869 \times 10^{-1}$	$1.899524010577561 \times 10^{-2}$

On average, the algorithm described above performs only one step of the continuation scheme. On every step of the continuation scheme it does about seven steps of Newton's iteration with around three step-size adjustments per each iteration.

The functions $G_n e^{im\phi}$ form a complete, orthonormal basis for functions on the spherical cap, hence we need to compute double integrals in colatitudinal and longitudinal variables. The procedure described above produces the nodes and weights for the colatitudinal variable. Integration with respect to the longitudinal variable can be done by a Gaussian quadrature as well, so the final quadrature rule is

$$\int_0^{2\pi} \int_b^1 f(x, \phi) dx d\phi = \sum_{i=1}^{n_\phi} w_i^\phi \left(\sum_{k=1}^n f(x_k, \phi_i) w_k^x \right),$$

Table 2. Nodes and weights of a $n = 5$ -, 10 -, 20 -point generalized Gaussian quadrature; $L = 3$; $m = 1$; $b = 0$; $W(x) = \sqrt{x - b}$.

N	Nodes x_i	Weights w_i
5	$5.171\,882\,969\,714\,534 \times 10^{-2}$	$2.341\,505\,038\,088\,086 \times 10^{-2}$
	$2.031\,623\,105\,912\,188 \times 10^{-1}$	$8.884\,898\,607\,298\,555 \times 10^{-2}$
	$4.356\,411\,421\,025\,461 \times 10^{-1}$	$1.714\,654\,253\,019\,344 \times 10^{-1}$
	$7.002\,764\,501\,137\,072 \times 10^{-1}$	$2.137\,749\,099\,088\,692 \times 10^{-1}$
	$9.160\,789\,294\,435\,172 \times 10^{-1}$	$1.544\,365\,124\,946\,195 \times 10^{-1}$
10	$1.470\,060\,286\,401\,878 \times 10^{-2}$	$3.562\,626\,303\,314\,735 \times 10^{-3}$
	$5.868\,587\,471\,593\,844 \times 10^{-2}$	$1.417\,300\,251\,017\,768 \times 10^{-2}$
	$1.314\,023\,014\,034\,694 \times 10^{-1}$	$3.141\,851\,068\,881\,919 \times 10^{-2}$
	$2.312\,259\,684\,709\,282 \times 10^{-1}$	$5.404\,249\,071\,427\,261 \times 10^{-2}$
	$3.545\,272\,566\,781\,712 \times 10^{-1}$	$7.926\,234\,023\,750\,671 \times 10^{-2}$
	$4.946\,547\,898\,729\,779 \times 10^{-1}$	$1.022\,669\,162\,191\,473 \times 10^{-1}$
	$6.412\,444\,375\,669\,494 \times 10^{-1}$	$1.164\,567\,246\,852\,908 \times 10^{-1}$
	$7.803\,824\,426\,328\,084 \times 10^{-1}$	$1.149\,571\,587\,384\,354 \times 10^{-1}$
	$8.960\,668\,473\,279\,682 \times 10^{-1}$	$9.335\,975\,293\,941\,803 \times 10^{-2}$
	$9.729\,912\,080\,571\,931 \times 10^{-1}$	$5.258\,717\,715\,892\,774 \times 10^{-2}$
20	$3.959\,614\,057\,989\,759 \times 10^{-3}$	$4.979\,655\,538\,890\,171 \times 10^{-4}$
	$1.582\,745\,120\,856\,827 \times 10^{-2}$	$1.990\,269\,851\,064\,446 \times 10^{-3}$
	$3.559\,214\,977\,062\,290 \times 10^{-2}$	$4.471\,865\,425\,313\,305 \times 10^{-3}$
	$6.321\,587\,478\,637\,355 \times 10^{-2}$	$7.927\,664\,681\,909\,845 \times 10^{-3}$
	$9.861\,524\,371\,928\,763 \times 10^{-2}$	$1.232\,329\,088\,245\,827 \times 10^{-2}$
	$1.416\,316\,287\,947\,248 \times 10^{-1}$	$1.759\,239\,754\,490\,365 \times 10^{-2}$
	$1.919\,943\,624\,001\,203 \times 10^{-1}$	$2.362\,155\,104\,951\,021 \times 10^{-2}$
	$2.492\,785\,459\,883\,913 \times 10^{-1}$	$3.023\,415\,310\,220\,783 \times 10^{-2}$
	$3.128\,601\,705\,648\,356 \times 10^{-1}$	$3.717\,573\,173\,420\,895 \times 10^{-2}$
	$3.818\,724\,965\,396\,798 \times 10^{-1}$	$4.410\,387\,041\,811\,009 \times 10^{-2}$
	$4.551\,689\,741\,193\,452 \times 10^{-1}$	$5.058\,684\,026\,339\,364 \times 10^{-2}$
	$5.312\,991\,690\,114\,278 \times 10^{-1}$	$5.611\,528\,812\,727\,251 \times 10^{-2}$
	$6.085\,048\,417\,924\,119 \times 10^{-1}$	$6.013\,067\,280\,364\,542 \times 10^{-2}$
	$6.847\,430\,949\,754\,737 \times 10^{-1}$	$6.207\,213\,941\,364\,178 \times 10^{-2}$
	$7.577\,419\,536\,990\,212 \times 10^{-1}$	$6.144\,003\,961\,120\,213 \times 10^{-2}$
$8.250\,906\,475\,975\,498 \times 10^{-1}$	$5.786\,968\,919\,983\,928 \times 10^{-2}$	
$8.843\,622\,938\,401\,178 \times 10^{-1}$	$5.120\,418\,486\,594\,962 \times 10^{-2}$	
$9.332\,611\,653\,960\,123 \times 10^{-1}$	$4.155\,170\,611\,239\,503 \times 10^{-2}$	
$9.697\,812\,394\,998\,446 \times 10^{-1}$	$2.931\,237\,128\,126\,203 \times 10^{-2}$	
$9.923\,585\,381\,315\,836 \times 10^{-1}$	$1.516\,351\,454\,911\,496 \times 10^{-2}$	

where nodes and weights x_k, w_k^x are products of the scheme presented before, and nodes and weights ϕ_i, w_i^ϕ are those that correspond to the quadrature rule for longitudinal variable with n_ϕ being the desired number of nodes.

Figures 4 and 5 show the nodes of the 40-point and 10-point quadratures on the sphere, where $L = 1, m = 1, b = -1/2$ and $L = 5, m = 2, b = 1/2$ correspondingly. The nodes tend to concentrate more at the bottom of the spherical cap rather than around the pole. A consequence of this fact is that the quadrature scheme described in this section does not produce the familiar ‘north pole oversampling’ problem.

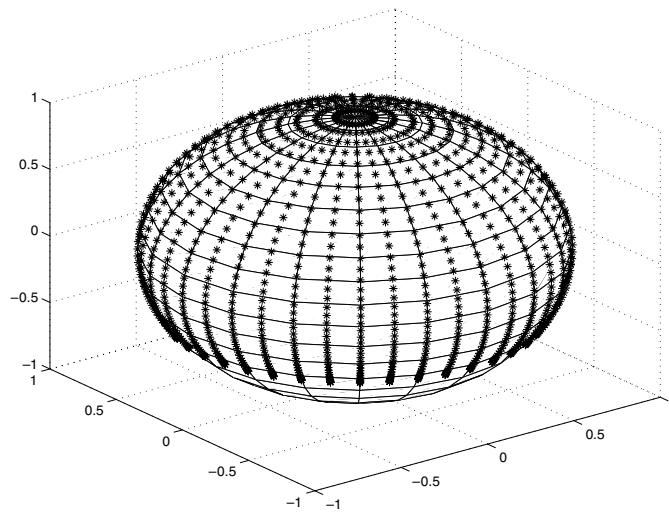


Figure 4. Nodes for 40-point quadrature, $b = 1/2$, $L = 1$, $m = 1$.

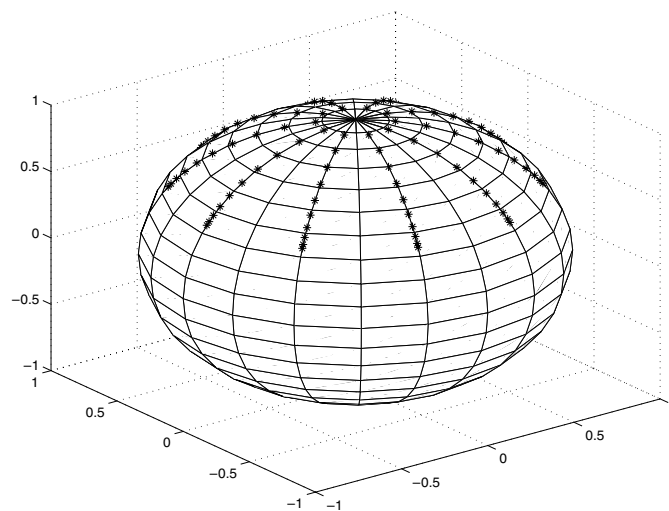


Figure 5. Nodes for 10-point quadrature, $b = 1/2$, $L = 5$, $m = 2$.

3. Conclusions

An efficient numerical scheme for evaluating a basis for the set of singular vectors of a time-band-time-limiting operator for certain regions on the surface has been presented. The basis functions form an orthonormal set, as well as an extended Chebyshev system, which guarantees the existence of a generalized Gaussian quadrature rule. An algorithm for computing weights and nodes for such a quadrature rule is presented. The nodes produced tend to concentrate not at the pole of the sphere, but at the bottom of the spherical cap, not exhibiting a common problematic effect called ‘north pole oversampling’.

Acknowledgments

The author is very grateful to Professor F A Grünbaum for many invaluable discussions. The author would also like to thank Professor W Kahan for a discussion on the Gaussian quadrature rule and for providing very useful references on the subject.

References

- [1] Grünbaum F A, Longhi L and Perlstadt M 1982 Differential operators commuting with finite convolution integral operators: some non-Abelian examples *SIAM J. Appl. Math.* **42** 941–55
- [2] Slepian D and Pollak H O 1961 Prolate spheroidal wave functions, Fourier analysis and uncertainty, I *Bell Syst. Tech. J.* **40** 43–64
- [3] Landau H J and Pollak H O 1961 Prolate spheroidal wave functions, Fourier analysis and uncertainty, II *Bell Syst. Tech. J.* **40** 65–84
- [4] Landau H J and Pollak H O 1962 Prolate spheroidal wave functions, Fourier analysis and uncertainty, III *Bell Syst. Tech. J.* **41** 1295–336
- [5] Slepian D 1964 Prolate spheroidal wave functions, Fourier analysis and uncertainty, IV *Bell Syst. Tech. J.* **43** 3009–58
- [6] Slepian D 1983 Some comments on Fourier analysis, uncertainty and modeling *SIAM Rev.* **25** 379–93
- [7] Slepian D 1976 On bandwidth *Proc. IEEE* **63** 292–300
- [8] Slepian D 1978 Prolate spheroidal wave functions, Fourier analysis and uncertainty, V *Bell Syst. Tech. J.* **57** 1371–430
- [9] Shannon C E 1948 A mathematical theory of communication *Bell Syst. Tech. J.* **27** 379–423
Shannon C E 1948 *Bell Syst. Tech. J.* **27** 623–56
- [10] Xiao H, Rokhlin V and Yarvin N 2001 Prolate spheroidal wave functions, quadrature, and interpolation *Inverse Problems* **17** 805–38
- [11] Albertella A, Sanso F and Sneeuw N 1999 Band-limited functions on a bounded spherical domain: the Slepian problem on the sphere *J. Geodesy* **73** 436–47
- [12] Grünbaum F A and Miranian L 2001 The magic of the prolate spheroidal functions in various setups *Proc. SPIE* **4478** 151–61
- [13] Hwang Ch 1990 Spectral analysis using orthonormal functions with a case study on the sea surface topography *Geophys. J. Int.* **115** 1148–60
- [14] Albertella A and Sneeuw N 2000 The analysis of radiometric data with Slepian functions *Phys. Chem. Earth A* **25** 667–72
- [15] Cheng H, Rokhlin V and Yarvin N 1999 Nonlinear optimization, quadrature, and interpolation *SIAM. J. Optim.* **9** 901–23
- [16] Karlin S and Studden W 1966 *Tchebyshev Systems with Applications in Analysis and Statistics* (New York: Wiley-Interscience)