Theoretical and Applied Informatics
ISSN 1896–5334
Vol.24 (2012), no. 1
pp. 67–94
DOI: 10.2478/v10179-012-0004-5

Approximation of values of prolate spheroidal wave function

MICHAL CHOLEWA

Institute of Theoretical and Applied Informatics
Polish Academy of Science
ul. Bałtycka 5, Gliwice, Poland
mcholewa@iitis.pl

Received 11 November 2011, Revised 28 November 2011, Accepted 26 December 2011.

Abstract: In this article we analyse various methods of value approximation for Prolate Spheroidal Wave Functions (PSWF). As PSWFs are not given by explicit formula, for any application their values need to be calculated based on their properties and connection to other functions. We will focus on three approaches – Legendre polynomials, Eigenvalues of Matrix Operators and Hermite functions. We then create an implementation and test its effectiveness by using is as a base for bandlimited signal approximation algorithm.

Keywords: Prolate Spheroidal Wave Functions,

1. Introduction

Prolate spheroidal wave functions create an orthonormal base for space of bandlimited signals. Approximation algorithm for band-limited signals based on prolate base coordinates has been proven optimal (in case where both error and jitter are equal to 0) and resilient to both error and jitter in pessimistic case (in [1]). That set of properties make prolate spheroidal wave functions a good option for approximation algorithms. However, for that an integral operator is needed – and for that operator to be applicable, we need to be able to know the value of PSWF in given x. As prolate spheroidal wave functions are not given by explicit formula, an approximation method is needed. There exist many methods of approximation PSWF value at given x. Natural method utilizes well known connection between PSWFs and Legendre polynomials, but that technique is not the only one. Some of them use Bessel functions ([3], [8]) and there are still pre-prepared value tables ([2], [4]) in use, event though they have a major vice – the the user has to use the same set of parameters that was used to generate the tables.

In [7] a solution based on eigenvalues of matrix operators, while in [9] authors base their approximation on Hermite functions.

In this article, we will go through three different method of PSWF approximation, then we will show and test an implementation of one of them.

This article is organized as follows: in section 2 we will conduct a short introduction concerning Prolate Spheroidal Wave Functions, in sections 3-5 we will focus on different approaches to the problem. Finally in section 6 we show and test implementation of PSWF approximation algorithm.

2. Prolate spheroidal wave functions (PSWFs)

Prolate Spheroidal Wave functions as an orthonormal base of bandlimited signal space give many possibilities for construction of approximation algorithms. Their properties are widely discussed in [6], [10] and [3], while in [5] Lindquist and Wager give practical application for fMRI signals.

In this section we will state only basics of their characteristic needed for purpose of approximation algorithms.

It is known that for every positive number c the values of parameter κ, so that differential equation

$$\forall_{t\in(-1,1)}(1-t^2)u''(t) - 2tu'(t) + (\kappa - c^2t^2)u(t) = 0, \tag{1}$$

has non-zero solution, can be ordered to form increasing sequence

$$0 < \kappa_0(c) < \kappa_1(c) < \kappa_2(c) < \dots$$

moreover, for $\kappa = \kappa_k(c)$ there exists an unique function $S_k(c,\cdot) : [-1,1] \leftarrow \mathbb{R}$ satisfying 1 so that $S_k(c,0) = P_k(0)$, where P_k is k-th Legendre polynomial, $k = 0, 1, 2, \dots$.

Functions $S_k(c,\cdot)$ have the following properties:

1. Each function $S_k(c,t)$ is continously dependant on c and for arbitrary c can be expanded to entire function of $t \in \mathbb{C}$.
2. Functions $S_k(c,\cdot)$ are orthogonal in $[-1,1]$ and complete in space $L_2(-1,1)$.
3. Each function $S_k(c,\cdot)$ has exactly k simple zeroes on $(-1,1)$.
4. Functions $S_k(c,\cdot)$ satisfy eigenequations:

$$\int_{-1}^{1} e^{icst} S_k(c,s)ds = 2i^k \alpha_k(c) S_k(c,t)$$

and

$$\int_{-1}^{1} \frac{\sin c(t-s)}{\pi(t-s)} S_k(c,s)ds = \lambda_k S_k(c,t)$$

where $\lambda_k = \lambda_k(c) = \frac{2c}{\pi}|\alpha_k(c)|^2$ and $\lambda_k(c) \searrow 0$ with $k \to \infty$.

Knowing this, we put

$$c = \Omega_0 \tau$$

and define Prolate Spheroidal Wave Functions $\psi_k : [-\tau, \tau] \to \mathbb{R}$ as follows:

$$\psi_k(t) = \lambda_k(c)^{1/2} \left(\int_{-1}^{1} S_k(c,s)^2 ds \right)^{-1/2} S_k(c, t/\tau) \tag{2}$$

having them defined this way, we can present $(1) - (4)$ for ψ_k as

(A) Each function $\psi_k(t)$ depends continuously of Ω_0 and τ, and for arbitrary Ω_0 and τ can be expanded to entire function $t \in \mathcal{C}$;

(B) Functions $\phi_k(t) = \lambda_k^{-1/2}\psi_k$ are orthogonal and complete in $L_2(-\tau, \tau)$;

(C) Each function ψ_k has exactly k simple zeroes

$$\xi_{k,1}, \xi_{k,2}, \ldots, \xi_{k,k}$$

on $(-\tau, \tau)$;

(D) Functions ψ_k satisfy eigenequations

$$\int_{-\tau}^{\tau} e^{\frac{i\Omega_0 ts}{\tau}} \psi_k(s) ds = 2i^k \tau \alpha_k \psi_k(t)$$

and

$$\int_{-\tau}^{\tau} \frac{\sin \Omega_0 (t-s)}{\pi(t-s)} \psi_k(s) ds = \lambda_k \psi_k(t)$$

where $\alpha_k = \alpha_k(\Omega_0\tau), \lambda_k = \lambda_k(\Omega_0\tau)$ and $\lambda_k \searrow 0$ with $k \to \infty$.

Substituting $s = \tau\Omega_0^{-1}\omega$ in first equation (D), we get

$$\psi_k(t) = \int_{-\Omega_0}^{\Omega_0} e^{i\omega t} X_k(\omega) d\omega \tag{3}$$

where $X_k(\omega) = (2i^k\Omega_0\alpha_k)^{-1}\psi_k(\tau\Omega_0^{-1}\omega)$. using second equation (D) and the fact, that

$$\int_{-\infty}^{\infty} \frac{\sin \Omega_0(t-s)}{\pi(t-s)} f(t) dt = f(s), \forall_{f \in B(\Omega_0, \tau)}$$

we get

$$(\psi_k, \psi_l) = \int_{-\infty}^{\infty} \psi_k(t) \psi_l(t) dt =$$

$$\frac{1}{\lambda_k}\int_{-\tau}^{\tau}\left(\int_{-\infty}^{\infty}\frac{\sin\Omega_0(t-s)}{\pi(t-s)}\psi_l(t)dt\right)\psi_k(s)ds = \tag{4}$$
$$\frac{1}{\lambda_k}\int_{-\tau}^{\tau}\psi_l(s)\psi_k(s)ds = \frac{1}{\lambda_k}(\psi_l,\psi_k)$$

Where $(\cdot,\cdot)$ i $\langle\cdot,\cdot\rangle$ are dot products respectively in $L_2(-\infty,\infty)$ and $L_2(-\tau,\tau)$. Equivalnces 2 i 3, together with (B) give us:

(E) Each function $\psi_k, k = 0, 1, \ldots$ is of energy 1 and is bandlimited to Ω_0. Moreover, set $\{\psi_k\}_{k=0}^{\infty}$ is orthonormal in $L_2(-\infty,\infty)$ and complete in $B(\Omega_0,\tau)$.

Above properties make PSWFs an effective tool to approximate bandlimited signals.

From (B) and (E) we easily can get representation for $B(\Omega_0,\tau)$:

(F) $B(\Omega_0,\tau) = \left\{f \in L_2(-\tau,\tau) : \Sigma_{k=0}^{\infty}\frac{|\langle f,\phi_k\rangle|}{\lambda_k} < \infty\right\}$

Among functions in $B(\Omega_0,\tau)$, orthogonal to $\psi_0,\psi_1,\ldots,\psi_{k-1}$, function ψ_k has highest energy concentration on $(-\tau,\tau)$, equal to $\lambda_k(c)$, i.e.

$$\lambda_k(c) = \sup\left\{\frac{\|f\|_{2,\tau}^2}{\|f\|_{2,\infty}^2} : f \in B(\Omega_0), (f,\psi_j) = 0, j = 0,1,\ldots,k-1\right\}. \tag{5}$$

Eigenvalues $\lambda_k(c)$ satisfy following inequalities:

(G) $\lambda_{[2c/\pi]-1} \geq 1/2, \lambda_{[2c/\pi]+1} \leq 1/2$

oraz

(H) $e^{-1/12}\frac{\pi c}{2I_k(k+1/2)}\left(\frac{c}{2\pi k}\right)^{2k} < \lambda_k(c) < \frac{2c}{\pi^2 k^2}\left(\frac{ec}{2k}\right)^2 k,$
where $I_k = \int_{-\pi/2}^{\infty}\left(\frac{\sin x}{x}\right)^{2k}dx$ and $k \geq 2c/\pi$

3. Legendre polynomials

This approach utilizes natural link between PSWF and Legendre polynomials, stated in definition as

$$S_k(c,0) = P_k(0), \tag{6}$$

where $P_k(0)$ is k-th Legendre polynomial. Legendre polynomials are orthogonal base of analytical functions, thus any PSWF can be expanded as

$$\psi_n = \sum_{i=0}^{\infty}\alpha_i^n\overline{P_i}. \tag{7}$$

where $\overline{P_i}$ is normalized Legendre polynomial. This approach reduces problem of approximating PSWF values to computation of value of sequence of polynomials.

3.1. Preliminaries

Definition 1 *Let $x \in \mathbb{R}$ and let $n \in \mathbb{N}$. By Legendre polynomials we will understand functions functions P_n given as:*

$$P_{n+1}(x) = \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x) \tag{8}$$

with $P_0(x) = 1$ and $P_1(x) = x$.

Above defined Legendre polynomials have the following properties

Fact 1 *Let P_k be the k-th Legendre polynomial for $k = 0, 1, \ldots$. Then:*

1. $$(1-x^2)\frac{d^2 P_k(x)}{dx^2} - 2x\frac{dP_k(x)}{dx} + k(k+1)P_k(x) = 0 \tag{9}$$
2. $P_k(1) = 1$
3. $$\int_{-1}^{1} P_k(x) P_j(x) dx = 0 \tag{10}$$

   for $i \neq j$.

Even though as above fact states, Legendre polynomials are orthogonal, on $\langle -1, 1 \rangle$, they are not orthonormal. To be exact

$$\int_{-1}^{1} (P_k(x))^2 dx = \frac{1}{n + 1/2} \tag{11}$$

Normalizing them we get *normalized Legendre polynomials $\overline{P_n}$* defined as:

$$\overline{P_n}(x) = P_n(x) \cdot \sqrt{n + 1/2} \tag{12}$$

We will now prove the following lemma:

Lemma 1 *For every integer $k \geq n$,*

$$\left| \int_{-1}^{1} x_k \overline{P_n}(x) dx \right| < \sqrt{\frac{2}{2k+1}} \tag{13}$$

For every integer $0 \leq k < n$

$$\left| \int_{-1}^{1} x_k \overline{P_n}(x) dx \right| = 0 \tag{14}$$

Proof:

We will begin with second part of the lemma.

Note, that $deg(\overline{P_k}) = k$ and that Legendre polynomials are linearly independent set of polynomials with real coefficients. That means, that $(\overline{P_0}, \overline{P_1}, \ldots, \overline{P_k})$ is a base of linear space of polynomials with degree not greater than k. Hence

$$x_k = \sum_{i=0}^{k} \alpha_i \overline{P_i}. \tag{15}$$

What follows, is

$$\left| \int_{-1}^{1} x_k \overline{P_n}(x) dx \right| = \left| \int_{-1}^{1} \sum_{i=0}^{k} \alpha_i \overline{P_i P_n}(x) dx \right| = \left| \sum_{i=0}^{k} \alpha_i \int_{-1}^{1} \overline{P_i P_n}(x) dx \right| = 0 \tag{16}$$

since $\overline{P_n}, n = 0, 1, \ldots$ are orthogonal.

Now we move to first part of the lemma.

From Cauchy-Schwartz inequality we have

$$\begin{aligned} \left| \int_{-1}^{1} x_k \overline{P_n}(x) dx \right| &= |\langle x_k, P_n \rangle| \leq \sqrt{\|x_k\| \|P_n\|} = \\ &= \int_{-1}^{1} x^k \cdot x^k dx = \sqrt{\tfrac{2}{2k+1}} \end{aligned} \tag{17}$$

QED

Knowing, that PSWFs $\psi_k, k = 0, 1, \ldots$ are analytical functions in $\mathbb{C}$, on interval $\langle -1, 1 \rangle$, we can expand each ψ_k into Legendre sequence of the form

$$\psi_j(x) = \sum_{k=0}^{\infty} \beta_k \overline{P_k}(x) \tag{18}$$

with coefficients β_k fast converging to 0, as is stated in lemma below.

Lemma 2 *Let $\overline{P_n}(x)$ be n-th normalized Legendre polynomial and $a \in \mathbb{R}$. Then:*

$$\int_{-1}^{1} e^{iax} \overline{P_n}(x) dx = \sum_{k=k_0}^{\infty} \alpha_k \int_{-1}^{1} x^{2k} \overline{P_n}(x) dx + i \sum_{k=k_0}^{\infty} \beta_k \int_{-1}^{1} x^{2k+1} \overline{P_n}(x) dx \tag{19}$$

where

$$\alpha_k = (-1)^k \frac{a^{2k}}{(2k)!} \tag{20}$$

$$\beta_k = (-1)^k \frac{a^{2k+1}}{(2k+1)!} \tag{21}$$

$$k_0 = \lfloor n/2 \rfloor \tag{22}$$

where $\lfloor \cdot \rfloor$ *is an integer part. Moreover for every integer* $m \geq \lfloor e \cdot |a| \rfloor + 1$,

$$\left| \int_{-1}^{1} e^{iax} \overline{P_n}(x)dx - \sum_{k=k_0}^{m-1} \alpha_k \int_{-1}^{1} x^{2k} \overline{P_n}(x)dx - i \sum_{k=k_0}^{m-1} \beta_k \int_{-1}^{1} x^{2k+1} \overline{P_n}(x) \right| \leq \left(\frac{1}{2}\right)^{2m}. \tag{23}$$

In particular, for

$$n \geq 2(\lfloor e \cdot |a| \rfloor + 1) \tag{24}$$

it gives

$$\left| \int_{-1}^{1} e^{iax} \overline{P_n}(x)dx \right| < \left(\frac{1}{2}\right)^{n-1} \tag{25}$$

Proof (after [8]):

The formula (19) is comes directly from lemma 1 and Taylor expansion of e^{iax}. To prove (23), let us assume integer

$$m \geq \lfloor e \cdot |a| \rfloor + 1 \tag{26}$$

and let

$$R_m = \sum_{k=m}^{\infty} \alpha_k \int_{-1}^{1} x^{2k} \overline{P_n}(x)dx + i \sum_{k=m}^{\infty} \beta_k \int_{-1}^{1} x^{2k+1} \overline{P_n}(x)dx. \tag{27}$$

From lemma 1 and triangle inequality, we get

$$|R_m| \leq \sum_{k=2m}^{\infty} \left(\frac{|a|^k}{k!} \cdot \sqrt{\frac{2}{2k+1}} \right) < \sum_{k=2m}^{\infty} \frac{|a|^k}{k!} \tag{28}$$

From (26) we have

$$\frac{|a|}{2m+k} < \frac{|a|}{2m} < \frac{1}{2e} < \frac{1}{2}. \tag{29}$$

For m given by (26) and integer $k > 0$, we can rewrite (28) as

$$|R_m| < \frac{|a|^{2m}}{(2m)!} \cdot \left(1 + \frac{1}{2} + \frac{1}{4} + \cdots\right) < 2\frac{|a|^{2m}}{(2m)!} \tag{30}$$

from where we obtain (23) using Stirling formula.

Finally we get (25), putting

$$m = \lfloor e \cdot |a| \rfloor + 1. \tag{31}$$

QED

Using above lemma we can estimate the quality of approximation of PSWFs using Legendre polynomials. Such estimation is given in following theorem proven by Xiao in [8] (without loss of generality, wce can assume that $\tau = 1$).

Theorem 1 *Let $\psi_m(x)$ be m-th prolate spheroidal wave function with band limit c and let $\overline{P_k}(x)$ be k-th normalized Legendre polynomial. Then, for every integer $m \geq 0$ and every real $c > 0$, if*

$$k \geq 2(\lfloor e \cdot c \rfloor + 1) \tag{32}$$

then

$$\left| \int_{-1}^{1} \psi_m(x)\overline{P_k}(x)dx \right| < \frac{1}{\lambda_m} \cdot \left(\frac{1}{2}\right)^{k-1}. \tag{33}$$

Moreover, for every $\epsilon > 0$, if

$$k \geq 2(\lfloor e \cdot c \rfloor + 1) + log_2(1/\epsilon) + log_2(1/\lambda_m), \tag{34}$$

then

$$\left| \int_{-1}^{1} \psi_m(x)\overline{P_k}(x)dx \right| < \epsilon. \tag{35}$$

Proof:

Obviously

$$\begin{aligned} \left| \int_{-1}^{1} \psi_m(x)\overline{P_k}(x)\mathrm{d}x \right| &= \frac{1}{|\lambda_m|} \left| \int_{-1}^{1} \psi_m(x) \left(\int_{-1}^{1} e^{\mathrm{i}cxt}\overline{P_k}(t)\mathrm{d}t \right) \mathrm{d}x \right| < \\ &< \frac{1}{|\lambda_m|} \int_{-1}^{1} |\psi_m(x)| \cdot \left| \int_{-1}^{1} e^{\mathrm{i}cxt}\overline{P_k}(t)\mathrm{d}t \right| \mathrm{d}x. \end{aligned} \tag{36}$$

Putting

$$a = cx \tag{37}$$

And knowing that prolate spheroidal wave functions have unit norm, so

$$\int_{-1}^{1} |\psi_m(x)| \mathrm{d}x \leq \sqrt{2}, \tag{38}$$

Together with formula 36 and lemma 2 it gives

$$\left| \int_{-1}^{1} \psi_m(x) \overline{P_k}(x) \mathrm{d}x \right| < \frac{1}{|\lambda_m|} \cdot \left(\frac{1}{2}\right)^{k-1} \int_{-1}^{1} |\psi_m(x)| \mathrm{d}x \leq \frac{1}{|\lambda_m|} \cdot \left(\frac{1}{2}\right)^{k-1/2}. \tag{39}$$

Now, putting $k \geq 2(\lfloor e \cdot c \rfloor + 1) + log_2(1/\epsilon) + log_2(1/\lambda_m)$ we get

$$\begin{aligned} \tfrac{1}{\lambda_m} \cdot \left(\tfrac{1}{2}\right)^{k-1} \geq \tfrac{1}{\lambda_m} \cdot \left(\tfrac{1}{2}\right)^{2(\lfloor e \cdot c \rfloor + 1) + log_2(1/\epsilon) + log_2(1/\lambda_m) - 1} = \\ = \tfrac{1}{\lambda_m} \cdot \left(\tfrac{1}{2}\right)^{2\lfloor e \cdot c \rfloor + 1} \cdot \epsilon \cdot \lambda_m \geq \epsilon \end{aligned} \tag{40}$$

Hence, for assumed k we have

$$\left| \int_{-1}^{1} \psi_m(x) \overline{P_k}(x) dx \right| < \epsilon. \tag{41}$$

QED

3.2. Approximation

In previous section, we provided boundaries fo coefficients α_k in expansion

$$\psi_j(x) = \sum_{k=0}^{\infty} \alpha_k^j P_k(x) \tag{42}$$

Now, we will focus on algorithm to calculate those coefficients.
Using (42) in differential equation

$$(1 - x^2)\psi''(x) - 2x\psi'(x) + (\chi_j - c^2 x^2)\psi(x) = 0, \tag{43}$$

and with recursive formula for Legendre polynomials

$$P_{n+1}(x) = \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x) \tag{44}$$

and differential equation

$$(1-x^2)\frac{\mathrm{d}^2 P_k(x)}{\mathrm{d}x^2} - 2x\frac{\mathrm{d}P_k(x)}{\mathrm{d}x} + k\cdot(k+1)P_k(x) = 0 \tag{45}$$

we get

$$\begin{aligned}\tfrac{(k+1)(k+2)}{(2k+3)(2k+5)}\cdot c^2\cdot\alpha^j_{k+2} + \Big(k(k+1) + \tfrac{2k(k+1)-1}{(2k+3)(2k-1)}\cdot c^2 - \chi_j\Big)\cdot\alpha^j_k \\ +\tfrac{k(k-1)}{(2k-3)(2k-1)}\cdot c^2\cdot\alpha^j_{k-2} = 0.\end{aligned} \tag{46}$$

We want, however, to expand prolate spheroidal weave functions in the base of normed Legendre polynomials

$$\psi_j(x) = \sum_{k=0}^{\infty}\beta^j_k\cdot\overline{P_k}(x) \tag{47}$$

where $\overline{P_k}(x) = P_k(x)\cdot\sqrt{k+1/2}$. Combining norming formula with recursion 46 we get similar equation but for β^j_k:

$$\begin{aligned}\tfrac{(k+1)(k+2)}{(2k+3)\sqrt{(2k+5)(2k+1)}}\cdot c^2\cdot\beta^j_{k+2} + \Big(k(k+1) + \tfrac{2k(k+1)-1}{(2k+3)(2k-1)}\cdot c^2 - \chi_j\Big)\cdot\beta^j_k \\ +\tfrac{k(k-1)}{(2k-1)\sqrt{(2k-3)(2k+1)}}\cdot c^2\cdot\beta^j_{k-2} = 0.\end{aligned} \tag{48}$$

We can now put above recursion in the matrix form. If for every $j =, 0, 1, \cdots$ we put β^j as

$$\beta^j = (\beta^j_0, \beta^j_1, \beta^j_2, \cdots), \tag{49}$$

then the following fact is true

Fact 2 *Coefficients χ_i are eigenvalues and vectors β^i - corresponding eigenvectors of the operator $l^2 \to l^2$ given as a symmetric matrix A, where*

$$A_{k,k} = k(k+1) + \frac{2k(k+1)-1}{(2k+3)(2k-1)}\cdot c^2 \tag{50}$$

$$A_{k,k+2} = \frac{(k+1)(k+2)}{(2k+3)\sqrt{(2k+5)(2k+1)}}\cdot c^2, \tag{51}$$

$$A_{k+2,k} = \frac{(k+1)(k+2)}{(2k+3)\sqrt{(2k+5)(2k+1)}}\cdot c^2, \tag{52}$$

for every $k = 0, 1, \ldots$, with all remaining elements of the matrix A equal to 0.

This fact allows us to rewrite recursion (48) as

$$(A - \chi_j \cdot I)(\beta^j) = 0, \tag{53}$$

Having constructed matrix A, we can formulate the algorithm of evaluating the value of j-th prolate spheroidal wave function $\psi_j(x)$ in the following manner:

- We generate first k rows and columns of matrix A, where k is given by inequality (34).
- We calculate eigenvectors $\{\beta^j\}$ and corresponding eigenvalues $\{\chi_j\}$ from A
- Obtained values of $\beta_0^j, \beta_1^j, \ldots$ can be used to get $\psi_j(x)$ using expansion $\psi_j(x) = \sum_{k=0}^{\infty} \beta_k^j \cdot \overline{P_k}(x)$

4. Eigenvalues of matrix operator

This approach to approximation of values of prolate spheroidal wave functions, proposed in [7], utilizes their characteristic as orthonormal system that maximizes energy concentration on given interval $[\tau, \tau]$, defining ϕ_0 as function of total energy $\|\phi_0\|^2 = 1$, maximizing the formula

$$\rho = \frac{\int_{-\tau}^{\tau} |f(t)|^2 dt}{\int_{-\infty}^{\infty} |f(t)|^2 dt}. \tag{54}$$

Each following function ϕ_k must also maximize (54), being at the same time orthogonal to $\{\phi_i\}_{i=0}^{k-1}$.

4.1. Introduction

Let f be a function with band limited by Ω. Using Shannon sampling theorem we can represent such functions as

$$f(t) - \sum_{n=-\infty}^{\infty} f_n S(t-n) \tag{55}$$

where S is a $sinc$ function, $S(t) = \frac{\sin(\Omega t)}{\Omega t}$, and f_n - coefficients in $\text{lin}\{S(\cdot - n)\}_{n=-\infty}^{\infty}$ (where by $\text{lin}\{f_i\}_{i=-\infty}^{\infty}$ we understand space spanned on functions f_i)- orthonormal base of B_Ω. Substituting (55) to (54), we get

$$\rho = \frac{\sum_{n=-\infty}^{\infty} f_n \sum_{k=-\infty}^{\infty} \overline{f_k} \int_{-\tau}^{\tau} S(t-n)S(t-k)dt}{\sum_{n=-\infty}^{\infty} f_n \sum_{k=-\infty}^{\infty} \overline{f_k} \int_{-\infty}^{\infty} S(t-n)S(t-k)dt}, \tag{56}$$

where obviously, following from Parseval equation for orthonormal base $\{S(\cdot - n)\}_{n=-\infty}^{\infty}$ of space B_Ω with product $l_2(B_\Omega)$ we get

$$\sum_{n=-\infty}^{\infty} f_n \sum_{k=-\infty}^{\infty} \overline{f_k} \int_{-\tau}^{\tau} S(t-n)S(t-k)dt = \sum_{n=-\infty}^{\infty} |f_n|^2. \tag{57}$$

Now, let us define infinite-dimensional, symmetric, real matrix

$$A_\tau = [a_\tau(n,k)] = \left[\int_{-\tau}^{\tau} S(t-n)S(t-k)dt \right], \tag{58}$$

which will allow us to show ρ as

$$\rho = \frac{\langle \mathrm{f}, A_\tau \mathrm{f} \rangle}{\langle \mathrm{f}, \mathrm{f} \rangle}, \tag{59}$$

where by f we understand a sequence $\{f(n)\}$, and dot product as product in $l_2(B_\Omega)$. We will use the same symbol A_τ to refer to operator on $l_2(B_\Omega)$ corresponding to matrix A_τ.

This matrix is positive defined, as dot product in numerator (59),

$$\langle \mathrm{f}, A_\tau \mathrm{f} \rangle = \int_{-\tau}^{\tau} |f(t)|^2 dt \tag{60}$$

is positive for non-zero f. This leads to following Lemma, stated in [7].

Lemma 3 *Operator given by matrix A_τ has limited Schmidt norm*

$$\|A_\tau\|_{HS} = \mathit{Tr}|A_\tau|^2. \tag{61}$$

Proof:

Indeed,we notice, that following Schwartz inequality we have

$$\left[\int_{-\tau}^{\tau} S(t-n)S(t-k)dt \right]^2 \leq \int_{-\tau}^{\tau} S^2(t-n)dt \int_{-\tau}^{\tau} S^2(t-k)dt. \tag{62}$$

Now, since

$$a_\tau(n,n) = \int_{-\tau}^{\tau} S^2(t-n)dt = \int_{-\tau}^{\tau} \frac{sin^2(\pi(t-n))}{\pi^2(t-n)^2} dt < \frac{2\tau}{\pi^2(n^2-\tau^2)} \tag{63}$$

for $n^2 > \tau^2$, we can see, that

$$\sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} |a_\tau(n,k)|^2 \leq \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} a_\tau(n,n) a_\tau(k,k) < \infty. \tag{64}$$

QED

We can now state the following fact

Fact 3 *Let A_τ be an operator on l^2 given by (58); A_τ is self-adjoint, compact and positive defined. Furthermore, its eigenvalues satisfied inequalities*

$$1 > \lambda_1 > \lambda_2 > \cdots > \lambda_n > \cdots > 0. \tag{65}$$

Next problem, which will be approached is maximalization of (59). It is a typical optimalization theory problem, solved by finding maximum eignevalue of A_τ and corresponding eigenvector. Solving this problem we obtain a sequence $\phi_0 = \{\phi_{0,k}\}$ and λ so that

$$A_\tau \phi_0 = \lambda \phi_0 \tag{66}$$

and $\|\phi\| = 1$ given norm on l^2. Also λ is maximum value of ρ. From there, we move to maximalization problem for (59) — function ϕ_0 is its solution. Indeed, since

$$\frac{\langle \phi_0, A_\tau \phi_0 \rangle}{\langle \phi_0, \phi_0 \rangle} = \frac{\langle \phi_0, \lambda \phi_0 \rangle}{\langle \phi_0, \phi_0 \rangle} = \lambda \tag{67}$$

where $\lambda = \max \rho$. Each following function ϕ_k will be a sequence $\phi_k = \{\phi_{k,n}\}$ orthogonal to all $\phi_i, i < k$ and maximizing (59). The sequence of functions $\{\phi_k(t)\}$ given

$$\phi_k(t) = \sum_{n=-\infty}^{\infty} \phi_{k,n} S(t-n) \tag{68}$$

will also be pairwise orthogonal in $L^2(\mathbb{R})$, as transformation $\{\phi_{k,n}\} \to \phi_k$ is an isometry between l^2 i $L^2(\mathbb{R})$. Functions $\{\phi_k(t)\}$ are orthonormal base of linear space B_Ω and are solutions to maximalization problem (54), which, according to the fact theta PSWFs are the only solution to this problem gives

Fact 4 *Prolate spheroidal wave functions $\phi_k(t)$ can be represented as*

$$\phi_k(t) = \sum_{n=-\infty}^{\infty} \phi_{k,n} S(t-n) \tag{69}$$

where $\{\phi_{k,n}\}$ are eigenfunctions of discrete operator A_τ on l^2 and

$$A_\tau = [a_\tau(n,k)] = \left[\int_{-\tau}^{\tau} S(tin) S(t-k) dt \right]. \tag{70}$$

Eigenvalues $\{\lambda_k\}$ are given as $\lambda_k = \int_{-\tau}^{\tau} |\phi_k(t)|^2 dt$

We can also observe, that PSWFs are obtained from $\phi_k(t)$ by normalizing λ_k.

4.2. Approximation

Of course in practical application of this method we cannot use the whole matrix A_τ as it is infinite-dimensional. The calculations utilizing the matrix and its eigenvectors and eigenvalues have to be conducted on finite approximations,the most natural of which is to truncate A_τ to limited dimension.

So, by A_τ^m we will understand the matrix given as

$$A_\tau^m = [a_\tau(n,k)], |n| \leq m, |k| \leq m. \tag{71}$$

We will use the same symbol to denote inifnite-dimensional matrix, being A_τ^m with all elements for $|n| > m$ or $|k| > m$ equal to 0. Inequality (63) allows us to find upper bound on elements of A_τ:

$$|a_\tau(n,k)|^2 \leq a_\tau(n,n)a_\tau(k,k) \leq \frac{2\tau}{\pi^2(n^2-\tau^2)} \frac{2\tau}{\pi^2(k^2-\tau^2)} \tag{72}$$

when $|k| > \tau$ and $|n| > \tau$. Furthermore if there exists $p, |p| > 1$ so that $|k| \geq p\tau$ and $|n| \geq p\tau$, its easy to see, that

$$|a_\tau(n,k)| \leq \frac{2}{\pi^2(p^2-1)\tau} \tag{73}$$

This leads to following bound

Fact 5 *Let A_τ be given by (58) and A_τ^m by (71) and let $p\tau > m$. Then, maximum difference between elements of A_τ and A_τ^m (infinite-dimensional) is*

$$e^m = |a_\tau(n,k) - a_\tau^m(n,k)| \leq \frac{2}{\pi^2(p^2-1)\tau}. \tag{74}$$

If we additionally assume, that

$$|p| > \sqrt{1 + \frac{2}{\pi\tau\epsilon}} \tag{75}$$

for any $\epsilon > 0$, then

$$e^m < \epsilon \tag{76}$$

Above fact shows, that increasing m, we can make e^m arbitrarly small and allows the use of A_τ^m as approximation of A_τ

5. Hermite functions

This approach to approximation is based on orthogonal base for band-limited signals, constructed from Hermite functions. Similarly to legendre polynomials, they are given by recursive equations, they make good base for approximating PSWF values for high Ω – band limit. This solution was introduced in [9], here we will sum up its main results.

5.1. Introduction

Definition 2 *Let $n \in \mathbb{N}$ and let $x \in \mathbb{R}$. By n-th Hermite polynomial we will understand:*

$$\begin{aligned} H_0(x) &= 1 \\ H_1(x) &= 2x \\ H_n(x) &= 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x) \end{aligned} \tag{77}$$

In further part of this section we will utilize some properties of Hermite polynomials, which we will state in following fact

Fact 6 *Let $n, m \in \mathbb{N}$ and let H_n be n-th Hermite polynomial. Then:*

A. H_n satisfies differential equation

$$H_n''(x) - 2xH_n'(x) + 2nH_n(x) = 0. \tag{78}$$

B. System $\{H_n\}$ is orthogonal on $\mathbb{R}$ with weight function e^{-x^2}, i.e for $n, m \in \mathbb{N}$

$$\int_{-\infty}^{\infty} e^{-x^2} H_n(x) H_m(x) dx = \sqrt{\pi} 2^n n! \delta_{nm}, \tag{79}$$

where δ_{nm} is Kroenecker delta.

C. Value of $H_n(x)$ is bounded by

$$|H_n(x)| < \frac{n! 2^{n/2} - [n/2]}{[n/2]!} e^{2|x|\sqrt{[n/2]}}. \tag{80}$$

Base consisting of Hermite polynomials is orthogonal, yet to become orthonormal, the polynomials must be scaled. We will be using *scaled Hermite polynomials $H_n^a(x)$*, where a is scale. We define them the following way

Definition 3 *Let $n \in \mathbb{N}$ and let $H_n(x)$ be n-th Hermite polynomial. Then, by scaled Hermite polynomial with scale $a \in \mathbb{R}$ we will understand polynomials $s \cdot H_n(x)$, orthonormal on $\mathbb{R}$ with weight function $e^{-a^2x^2}$, $a > 0$.*

$$\int_{-\infty}^{\infty} e^{-a^2x^2} H_n^a(x) H_m^a(x) dx = \delta_{m,n}. \tag{81}$$

So

$$H_n^a(x) = \frac{\sqrt{a}}{\pi^{1/4} \cdot 2^{n/2} \cdot (n!)^{1/2}} \cdot H_n(ax) \tag{82}$$

and (following (78))

$$\frac{1}{a^2} \cdot \frac{d^2 H_n^a(x)}{dx^2} - 2x \frac{dH_n^a(x)}{dx} + 2nH_n^a(x) = 0. \tag{83}$$

of course following above and (77) we can immediately get recursion between scaled Hermite polynomial as

$$\begin{aligned} H_0^a(x) &= \sqrt{a}\left(\tfrac{1}{\sqrt{\pi}}\right)^{1/2} \\ H_1^a(x) &= \sqrt{2a}\left(\tfrac{1}{\sqrt{\pi}}\right)^{1/2} ax \\ H_n^a(x) &= ax\sqrt{\tfrac{2}{n}}H_{n-1}^a(x) - \sqrt{\tfrac{n-1}{n}}H_{n-2}^a(x). \end{aligned} \tag{84}$$

For scaled hermite polynomials, following recursions are true:

Fact 7 *Let $s \neq 0$ and let $H_n^a(x)$ be n-th scaled Hermite polynomial with scale a. Then, for every $n \in \mathbb{N}$*

1.

$$xH_n^a(x) = \frac{1}{a}\sqrt{\frac{n+1}{2}}H_{n+1}^a(x) + \frac{1}{a}\sqrt{\frac{n}{2}}H_{n-1}^a(x), \tag{85}$$

2.

$$\begin{aligned} x^2H_n^a(x) = \tfrac{1}{a^2}\sqrt{\tfrac{n+1}{2}\tfrac{n+2}{2}}H_{n+2}^a(x) + \tfrac{1}{a^2}\left(n+\tfrac{1}{2}\right)H_n^a(x) + \\ +\tfrac{1}{a^2}\sqrt{\tfrac{n}{2}\tfrac{n-1}{2}}H_{n-2}^a(x), \end{aligned} \tag{86}$$

3.

$$\begin{aligned} x^4H_n^a(x) = \tfrac{1}{a^4}\sqrt{(n+1)(n+2)(n+3)(n+4)}H_{n+4}^a(x) \\ +\tfrac{1}{a^4}\left(n+\tfrac{3}{2}\right)\sqrt{(n+1)(n+2)}H_{n+2}^a(x) \\ +\tfrac{1}{a^4}\tfrac{3}{4}(1+2n+2n^2)H_n^a(x) \\ +\tfrac{1}{2a^4}\sqrt{(n-1)n}(2n-1)H_{n-2}^a(x) \\ +\tfrac{1}{4a^4}\sqrt{(n-3)(n-2)(n-1)n}H_{n+4}^a(x), \end{aligned} \tag{87}$$

with assumption that $H_n^a(x) \equiv 0$ for $n < 0$.

Now, we can define Hermite functions based on Hermite polynomials

Definition 4 *Let $a > 0, a \in \mathbb{R}$ and let H_n^a be n-th scaled Hermite polynomial with scale a. By Hermite functions $\phi_0^a, \phi_1^a, \phi_2^a, \ldots : \mathbb{R} \to \mathbb{R}$ we will understand functions given by*

$$\phi_n^a(x) = e^{-a^2x^2/2} \cdot H_n^a(x). \tag{88}$$

Hermite functions posses properties which make them useful in PSWF approximation. Some of them we can state in following fact.

Fact 8 *Let $\phi_0^a, \phi_1^a, \phi_2^a, \ldots : \mathbb{R} \to \mathbb{R}$ be Hermite functions. Then*

1. *Dla $n \geq 1$*
$$x\phi_n^a(x) = \frac{1}{a}\sqrt{\frac{n+1}{2}}\phi_{n+1}^a(x) + \frac{1}{a}\sqrt{\frac{n}{2}}\phi_{n-1}^a(x). \tag{89}$$

2. *Dla $m, n \geq 0$ zachodzi*
$$\int_{-\infty}^{\infty} \phi_m^a(x)\phi_n^a(x)dx = \delta_{mn} \tag{90}$$
where δ_{mn} is Kroenecker delta.

3. *For every $f \in l^2$*
$$f(x) = \sum_{n=0}^{\infty} \alpha_n \phi_n^a(x), \tag{91}$$
where α_n are given by formula
$$\alpha_n = \int_{-\infty}^{\infty} f(x)\phi_n^a(x)dx. \tag{92}$$
Furthermore, if f is even, $\alpha_{2n+1} = 0\forall_{n\geq 0}$, and if f is odd, $\alpha_{2n} = 0\forall_{n\geq 0}$. To expansion (101) we will refer to as Hermite expansion.

5.2. Approximation

Let G_c be a differential operator (for $c > 0$) given as

$$G_c(\psi)(x) = -(1-x^2)\psi''(x) + 2x\psi'(x) + c^2x^2\psi(x). \tag{93}$$

Operator G_c is linear, self-adjoint and positive defined. The following theorem comes directly as conclusion from fact (7).

Fact 9 *Let $a > 0, a \in \mathbb{R}$ and let $\{\phi_n^a\}$ be a sequence of Hermite functions. Than for any real and positive c and positive integer n*

$$G_c\phi_n^a(x) = \sum_{i=-2}^{2} d_{n,i} \cdot \phi_{n+2i}^a(x), \tag{94}$$

where

$$d_{n,0} = \frac{1}{4a^2}(-3a^2 + 2a^4 + 2c^2 - 2a^2n + 4a^4n + 4c^2n - 2a^2n^2), \tag{95}$$

$$d_{n,1} = -\frac{1}{2a^2}(a^2 - c)(a^2 + c)\sqrt{2 + 3n + n^2}, \tag{96}$$

$$d_{n,2} = \frac{1}{4}\sqrt{(3+n)(4+n)(2+3n+n^2)} \tag{97}$$

for $n \geq 0$*;*

$$d_{n,-1} = -\frac{1}{2a^2}(a^2 - c)(a^2 + c)\sqrt{-n + n^2} \tag{98}$$

for $n \geq 2$*;*

$$d_{n,-2} = \frac{1}{4}\sqrt{(-3+n)(-2+n)}\sqrt{-n + n^2} \tag{99}$$

for $n \geq 4$*. Moreover*

$$d_{3,-2} = d_{2,-2} = d_{1,-2} = d_{0,-2} = d_{1,-1} = d_{0,-1} = 0. \tag{100}$$

Now lets assume that ψ_m^c is eigenfunction of operator G_c. Of course for every $a > 0$ we can expand ψ_m^c into Hermite expansion as follows

$$\psi_m^c(x) = \sum_{n=0}^{\infty} \alpha_n^m \phi_n^a(x), \tag{101}$$

where α_n^m depend on a and c. The following theorem (after [9]) illustrates dependencies between them.

Theorem 2 *Let χ_m be m-th eigenvalue G_c and let ψ_m^c be corresponding eigenfunction. Moreover, let $\alpha_0^m, \alpha_1^m, \alpha_2^m, \ldots$ be Hermite expansion coefficients (101) ψ_m^c. Then, for $n \geq 0$,*

$$\begin{gathered} \tfrac{1}{4}\sqrt{-3+n}\sqrt{-2+n}\sqrt{-n+n^2}\cdot\alpha_{n-4}^m - \tfrac{1}{2a^2}(a^4 - c^2) \\ \cdot\sqrt{-n+n^2}\cdot\alpha_{n-2}^m - \Big(\chi_m - \tfrac{1}{4a^2}(-3a^2 + 2a^4 + 2c^2 - 2a^2 n \\ +4a^4 n + 4c^2 n - 2a^2 n^2)\Big)\cdot\alpha_n^m - \tfrac{1}{2a^2}(a^4 - c^2)\sqrt{2+3n+n^2} \\ \cdot\alpha_{n+2}^m + \tfrac{1}{4}\sqrt{3+n}\sqrt{4+n}\sqrt{2+3n+n^2}\cdot\alpha_{n+4}^m = 0. \end{gathered} \tag{102}$$

Proof:

Using G_c to both sides of (101) we get

$$G_c(\psi_m^c)(x) = G_c\left(\sum_{n=0}^{\infty}\alpha_n^m\phi_n^a(x)\right) = \sum_{n=0}^{\infty}\alpha_n^m G_c(\phi_n^a(x)). \tag{103}$$

From fact (9) we know, that

$$G_c(\phi_n^a(x)) = \sum_{i=-2}^{2} d_{n,i} \cdot \phi_{n+2i}^a(x) \tag{104}$$

for $n \geq 0$ and $d_{i,n}$ given in fact (9). Substituting above to (103) we get

$$G_c(\phi_n^a(x)) = \sum_{n=0}^{\infty} \alpha_n^m \sum_{i=-2}^{2} d_{n,i} \cdot \phi_{n+2i}^a(x) = \sum_{n=0}^{\infty} \left(\sum_{i=-2}^{2} \alpha_n^m d_{n,i} \right) \cdot \phi_{n+2i}^a(x). \tag{105}$$

On the other hand, since χ_m is eigenvalue of G_c and ψ_m^c – corresponding eigenfunction

$$G_c(\psi_m^c)(x) = \chi_m \cdot \sum_{n=0}^{\infty} \alpha_n^m \cdot \phi_n^a(x). \tag{106}$$

Comparing those two equalities we get the thesis.

QED.

Using inverse power method in Mathematica, those recurrence can be translated into following formula

$$\psi_m^c(x) = \sum_{i=0}^{\infty} \sum_{k=0}^{\infty} \frac{\alpha_{i,k}}{c^k} \cdot \phi_{m+4i}^{\sqrt{c}}(x) + \sum_{i=1}^{[m/4]} \sum_{k=1}^{\infty} \frac{\beta_{i,k}}{c^k} \cdot \phi_{m-4i}^{\sqrt{c}}(x) \tag{107}$$

where $\alpha_{i,k}$ and $\beta_{i,k}$ are functions of m.

Of course, it is in practice impossible to calculate the sum of infinite series, therefore a truncated series will be used

$$\psi_m^{c,n}(x) = \sum_{i=0}^{n} \sum_{k=0}^{n} \frac{\alpha_{i,k}}{c^k} \cdot \phi_{m+4i}^{\sqrt{c}}(x) + \sum_{i=1}^{[m/4]} \sum_{k=1}^{n} \frac{\beta_{i,k}}{c^k} \cdot \phi_{m-4i}^{\sqrt{c}}(x) \tag{108}$$

which – introducing notation $\alpha_i^n = \sum_{k=0}^{n} \frac{\alpha_{i,k}}{c^k}$ and $\beta_i^n = \sum_{k=0}^{n} \frac{\beta_{i,k}}{c^k}$ can be rephrased as

$$\psi_m^{c,n}(x) = \sum_{i=0}^{n} \alpha_i^n \cdot \phi_{m+4i}^{\sqrt{c}}(x) + \sum_{i=1}^{[m/4]} \beta_i^n \cdot \phi_{m-4i}^{\sqrt{c}}(x) \tag{109}$$

Naturally, the question arises, how accurate this approximation is. This problem is addressed in following fact which is proven in [9]

Fact 10 *Sequence $\psi_m^{c,n}$ is convergent to ψ_m^c uniformly in $\mathbb{R}$ as $c \to \infty$. Also*

$$\|\psi_m^{c,n}(x) - \psi_m^c(x)\|_{[-\infty,\infty]} = O\left(\frac{1}{c^{n+1}}\right) \tag{110}$$

Above theorem validates truncation as approximation method.

Use of Mathematica allows us also to approximate both α_i^n and β_i^n. For implementation in next section we decided to use $n = 5$ for our algorithm.

6. Implementation

Using the inverse power method allowed calculating also polynomial approximations of χ_m and $\psi_m^c(x)$ for given c (shown in of [9]). The following implementation of algorithm approximating values of PSWF for any given x is based on their results. We introduce it having in mind lack of such algorithms in literature, and then test it on sample function.

```
import numpy as np
import s c ipy as sp

def HermitePoly (n, x):
   if (n == 0):
      return 1.0
   X = np.  ar ray (range (n+1), dtype=f l o a t)
   X[0] = 1.0
   X[1] = 2.0 * x
   for i in range (2, n+1):
      X[i] = 2.0 * x * X[i-1] - 2.0 * (i-1.0)* X[i-2]
   return X[n]

def ScaledHermi tePoly (n, scale, x):
   HP=HermitePoly (n, scale * x)
   wsp = np.sqrt (scale) / (np.pi**(0.25)*(2**(n/2))*
   np.  sqrt (sp.  factorial(n)))
   return HP * wsp

def HermiteFunction (n, scale, x):
   SHP=ScaledHermi tePoly (n, scale, x)
   wsp = np.  power (np.  e, -(np.  power (scale*x, 2)/2))
   return SHP * wsp
def Alpha5 (i, c, m):
   wynik = 0
   if (i == 0):
     wynik += 1
    wynik -= (1/(2**10* c **2))*(12 + 22*m + 23*m**2
    + 2*m**3 + m**4)
    wynik -= (1/(2**11* c **3)) *(60 + 158*m + 115
     * m**2 + 80 * m**3 + 5 * m**4 + 2 * m**5)
    wynik -= ((1/(2**22 * c **4) )*(328032 + 891024*m
```

```
  + 1127140*m**2 + 476156 * m**3 + 247887*m**4
  + 11768*m**5 + 3918*m**6 - 4*m**7 - m**8))
  wynik -= ((1/(2**22* c **5)) * (993120 + 3161552*m
  + 3698884*m**2 + 3044356*m**3 + 874439*m**4
  + 363350*m**5 + 13566*m**6 + 3864*m**7 - 9*m**8
  - 2*m**9))
  return wynik
if (i == 1):
  wynik -= ((1/(2**5 * c)) *
  np.  sqrt (sp.  factorial (m+4)/ sp.  factorial (m)) *
  (1 + (1/(4* c)) * (5 + 2*m) + (1/(2**11* c **2)) *
  (4808 + 3470*m + 669*m**2 - 10 * m**3 - m**4) +
  (1/(2**13* c **3)) * (46840 + 46762*m + 16499*
  m**2 + 1920*m**3 - 71*m**4 - 6*m**5) + (1/(3
  *2**22* c **4)) *(212454624 + 263405280*m +
  128877012*m**2 + 29276108*m**3 + 2118049*m**4
  - 151072*m**5 - 1030*m**6 + 20*m**7 + m* * 8)))
  return wynik
if (i == 2):
  wynik += (1/(2**11 * c **2)) *
  np.  sqrt (sp.factorial(m+8)/sp.factorial(m)) * (1 +
  (1/(2* c)) * (7 + 2*m) + (1/(3*2**10* c **2)) *
  (3738 + 19698*m + 2833*m**2 - 18*m**3 - m**4)+
  (1/(3*2**9* c **3)) * (70716 + 52218*m + 13869
  *m**2 + 1291*m**3 - 21*m**4 - m**5))
  return wynik
if (i == 3):
  wynik -= (1/(3 * 2**16 * c **3)) *
  np.  sqrt (sp.factorial(m+ 12)/sp.factorial (m)) *(1 +
  4/(3* c)*(9 + 2*m) + 1/(2**12* c **2) * (154128 +
  64022*m + 7237*m**2 - 26*m**3 - m**4))
  return wynik
if (i == 4):
  wynik += (1/(3 * 2**23 * c **4)) *
  np.  sqrt (sp.factorial(m + 16)/sp.factorial(m)) *
  (1 + 1/ c * (11 + 2 * m))
  return wynik
if (i == 5):
  wynik -= (1/(15 * 2**28 * c **5)) *
```

```
    np. sqrt (sp. factorial (m+20)/sp. factorial (m))
    return wynik
   return wynik

def Beta5 (i, c,m):
   wynik = 0
   if (i == 1):
   wynik += ((1/(2**5 * c)) *
    np. sqrt (sp. factorial (m)/sp. factorial (m−4)) *
    (1 − 1/(4* c) * (3 − 2*m) + (1/(2**11* c **2)) *
    (2016− 2106*m + 693 * m**2 + 6 * m**3 − m**4) −
    (1/(2**13* c **3)) * (14592 − 19788*m + 10373*m**2
    − 2144*m**3 − 41*m**4 + 6*m**5) +
    1/(3*2**22* c **4) )* (50908320 − 84318336*m +
    55101860*m**2 − 19514436*m**3 + 2707329*m**4 +
    84528*m**5 − 11142*m**6 − 12*m**7 + m**8))
    return wynik
   if (i == 2):
    wynik += ((1/(2**11* c **2)) *
    np. sqrt (sp.factorial(m)/sp.factorial(m−8)) * (1 −
    1/(2* c) * (5−2*m)
    + (1/(3 * 2**10* c **2)) * (20460 − 13982*m +
    2881*m**2 + 14*m**3 − m**4) − (1/(3*2**9* c **3))
    * (31056 − 28432*m + 9880*m**2 − 1365*m**3 −
    16*m**4 + m* * 5)))
    return wynik
   if (i == 3):
    wynik += ( (1/(3*2**16* c **3)) *
    np. sqrt (sp.factorial(m)/sp.factorial(m−12)) * (1 −
    3/(4* c) * (7 − 2*m) + 1/(2**12* c **2) *
    (97368 − 49474*m + 7309*m**2 + 22*m**3 − m* * 4)))
    return wynik
   if (i == 4):
    wynik += (1/(3 * 2**23 * c **4) ) *
    np. sqrt (sp. factorial (m)/sp. factorial (m−16))
    * (1 − 1/ c * (9−2*m))
    return wynik
   if (i == 5):
    wynik += (1/(15*2**28* c **5)) *
    np. sqrt (sp. factorial (m)/sp. factorial (m−20))
```

```
        return wynik
    return wynik

def PSWFValue(n, x, c):
        wynik = 0
        scale = np.  sqrt (c)
        for i in range (6):
        wynik += Alpha5 (i, c, n) * HermiteFunction (n+4*i,
         scale, x)
        limit = np.minimum (5, n/4)
        for j in range (1, limit):
        wynik += Beta5 (j, c, n) * HermiteFunction (n - 4* j,
         scale, x)
        return wynik

def PSWFScaledValue (n, x, c, tau):
    return 1/np.  sqrt (tau) * PSWFValue(n, x/tau, c)
```

Based on the above implementation, it is easy to construct the approximation algorithm. To test it, we have applied it against a function g, where

$$g(x) = \begin{matrix} 20\sin(x), |x| \leq \pi \\ 0, |x| > \pi \end{matrix} \tag{111}$$

We have used increasing number of PSWFs to approximate g. The effects can be seen in 1.

Another function we can test our algorithm against will be

$$f(x) = \begin{matrix} 20, |x| \leq 0.4 \\ 200(|0.5 - |x||), 0.4 < |x| \leq 0.5 \\ 0, |x| > 0.5 \end{matrix} \tag{112}$$

Choice of f shows us how the algorithm behaves being applied to scaled unit signal. Figure 2 shows the results. As we can observe, approximation of f achieves high level of precision relatively fast. Above 40 PSWF approximation changes are very small and focus mainly around the points where the function f is not differentiable.

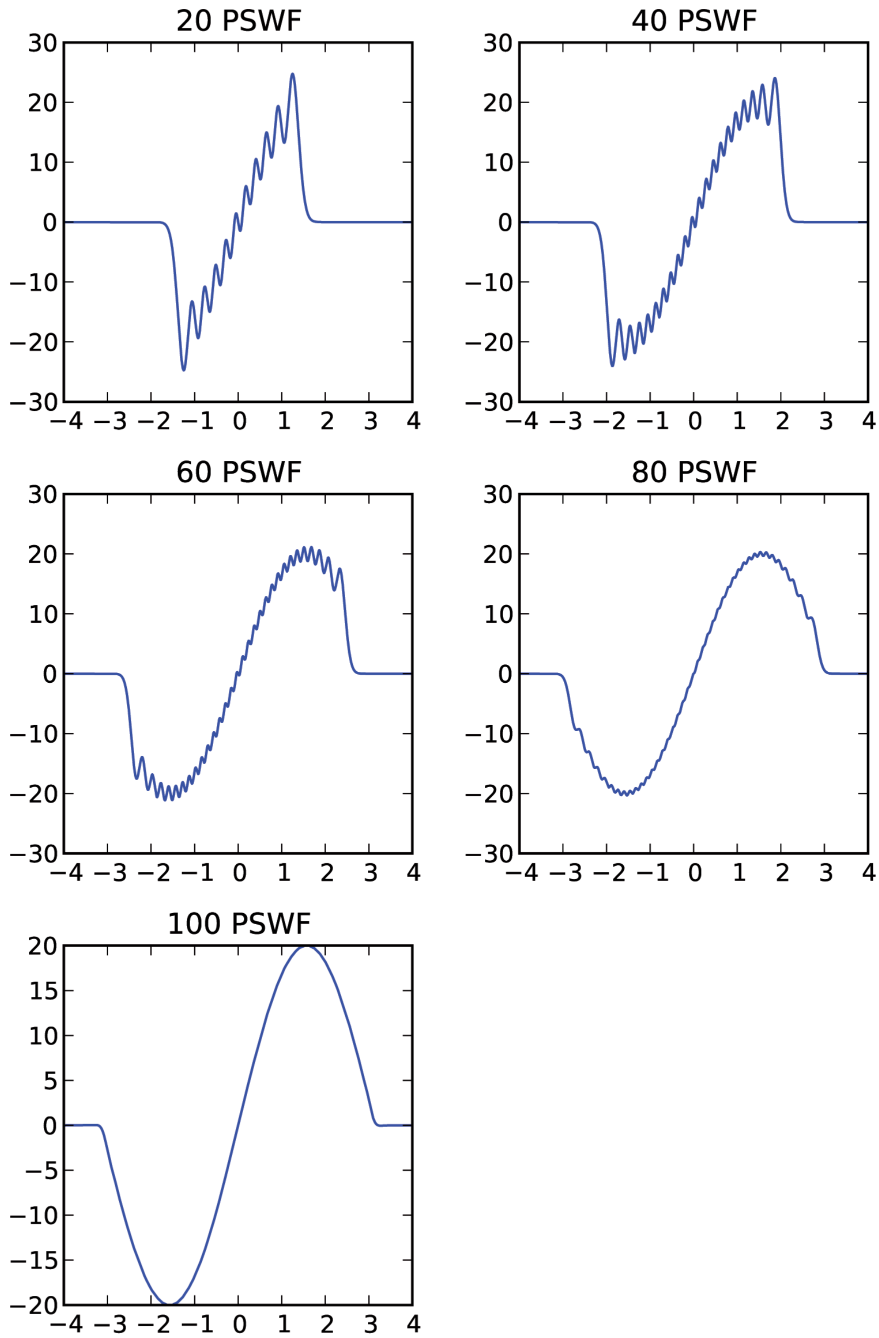

Fig. 1. Approximation of function g using increasing number of PSWF

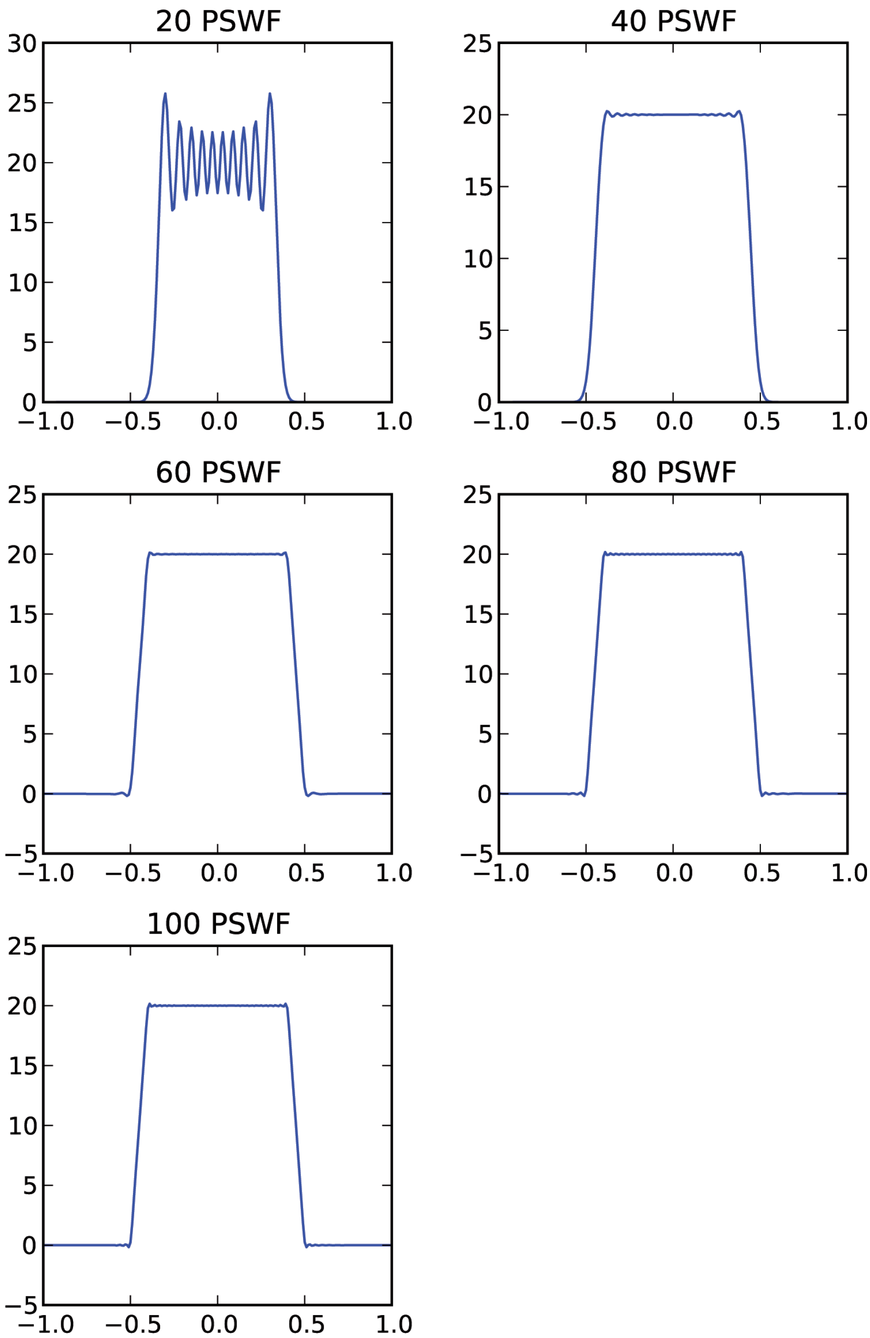

Fig. 2. Aproksymacja funkcji f przy pomocy rosnacej liczby czolowych funkcji kulistych (PSWF)

7. Conclusions

Since Prolate Spheroidal Wave Functions are an effective tool for approximating bandlimited signals and their definition does not allow any straightforward calculations, a method for estimating their values was an important problem in digital signal processing field. Three of them mentioned above utilize two different orthonormal basis of bandlimites space and eignevalues of matrix operator and all of them yield error with very low upper bound even for relatively low parameters, and therefore fast computations.

These three methods show that even though PSWFs are not explicitly represented, they can be approximated with high accuracy, which makes them valid tool for bandlimited signal processing.

References

1. D. Dąbrowska: *Skutki zaburzeń miejsc i wartości odczytu próbek funkcji w optymalnym odtwarzaniu. Czynić sprawiedliwość w miłości: Księga pamiątkowa od Uniwersytetu Kardynała Stefana Wyszyńskiego dla Jego Eminencji Józefa Kardynała Glempa w dwudziestą rocznicę posługi Prymasowskiej*, pages 424-490, 2001.

2. C. Flammer: *Spheroidal Wave Functions*, Stanford University Press, Stanford, CA, 1957.

3. H. Volkmer: *Spheroidal Wave Functions in Handbook of Mathematical Functions*, Nat. Bureau of StS. Applied Math Series, 2004.

4. S. Hanish, R.V. Baier, A.L. Van Buren, and B.J. King: *Tables of Radial Spheroidal Wave Functions*, Volume 1: Prolate, m = 0. Tach. Rep. NRL 7088, Naval Research Labolatory, 1970.

5. M.A. Lindquist and T.D.Wager: *Spatial smoothing in fmri using prolate spheroidal wave functions*. Human Brain mapping., 26:1276-1287, 2008.

6. D. Walter and X. Shen: *Wavelets based on prolate spheroidal wave functions*. Fourier Anal. Appls., 10:1-26, 2004.

7. G.G. Walter and T. Soleski: *A new, friendly method of computing prolate spheroidal wave functions and wavelets*. Appl. Comput. Harmon. Anal., 19(3):432-443, 2005.

8. H. Xiao: *Prolate spheroidal wavefunctions, quadrature and interpolation*. Inverse Problems, 17:805-838, 2001.

9. H. Xiao and V Rokhlin: *High-frequency asymptotic expansions for certain prolate spheroidal wave functions*. Journal of Fourier Analysis and Applications, 9:575-596, 2003.

10. H. Xiao, V. Rokhlin, and N. Yarvin: *Prolate spheroidal wavefunctions, quadrature and interpolation*. J. Inverse Problems, 17:805-838, 2001.

Aproksymacja wartości czołowych funkcji kulistych

Streszczenie

W artykule analizujemy różnorakie metody przybliżania wartości czołowych funkcji kulistych (Prolate Spheroidal Wave Functions – PSWF). Jako że funkcje te nie są zadane poprzez bezpośredni wzór, konieczne do zastosowań obliczenie ich wartości w zadanych punktach nie jest sprawą trywialną. Obecnie czyni się to w oparciu o ich związki z innymi funkcjami, łatwiej obliczalnymi. W artykule koncentrujemy się nad trzema podejściami – poprzez naturalny związek czołowych funkcji kulistych z wielomianami Legendre'a, przez funkcje Hermite'a oraz jako wartości własne operatorów macierzowych.

Następnie wskazujemy implementację przykładowego algorytmu obliczania wartości PSWF i pokazujemy jego działanie na przykładzie przybliżeń sygnałów o ograniczonym paśmie.