TECHNICAL REPORT
Explicit expressions for d-dimensional spherical surface harmonics by the program SSHY*

Thomas Wieder†

23 July 2009

Abstract

This technical report describes the Maple programm SSHY. By SSHY one can calculate either symbolically or numerically d-dimensional spherical surface harmonics of order n.

1 Introduction

This technical report describes the Maple 12 program SSHY. The purpose of SSHY is just to provide the user with explicit expressions for spherical surface harmonics of dimension $d \geq 3$. No new mathematical results are presented. All underlying mathematical formulas were taken from Bateman, Higher Transcendental Functions, Vol. II, [1].

2 Harmonic polynomials and Spherical Surface Harmonics

The following definitions, theorems and equations are taken from Bateman [1]. Thus, this chapter is a paraphrase of chapters 11.1 to 11.3 from Bateman.

Let $\vec{x}$ be a d-dimensional point vector in a d-dimensional euclidian space E^d. Harmonic polynomials in E^d are defined according to as follows: "A polynomial $H_n(\vec{x})$ of degree n in $x_1, x_2, \ldots, x_{p+2}$ which is homogeneous of degree n, so that $H_{\lambda n} = \lambda^n H_n(\vec{x})$, and satisfies Laplace's equation $\Delta H_n(\vec{x}) = 0$, is known as a *harmonic polynomial* of degree n."

It is possible to construct a complete set of linearly independent harmonic polynomials of degree n. The construction is outlined in [1] from where we copy the defining formula (3).

---

*Mathematics Subject Classifications: 33C55.

†Kassel, Germany, thomas.wieder@t-online.de

Let $m_0, \ldots, m_p$ be integers such that

$$N = m_0 \geq m_1 \geq \ldots \geq m_p \geq 0, \tag{1}$$

and let r_k be defined by

$$r_k = (x_{k+1}^2 + x_{k+2}^2 + \ldots + x_{p+2}^2)^{1/2} \tag{2}$$

where $k = 0, 1, \ldots, p$ and $r_0 = r$. Then

$$\begin{aligned} H(m_k, \pm m_p, \vec{r}) &= H(m_k, \pm m_p, x_1, \ldots, x_{p+2}) \qquad (3) \\ &= \left(\frac{x_{p+1}}{r_p} + i\frac{x_{p+2}}{r_p}\right)^{\pm m_p} r_p^{m_p} \prod_{k=0}^{p-1} r_k^{m_k - m_{k+1}} C_{m_k - m_{k+1}}^{m_{k+1}+1/2(p-l)}(x_{k+1}/r_k). \end{aligned}$$

Within E^d we have the hyper-sphere S^{d-1} of radius $\vec{r} = 1$. The restriction of $H_n(\vec{x})$ to S^{d-1} results in spherical surface harmonics $Y_n(\vec{x})$. [1] We will need hyperspherical polar coordinates $r, \theta_1, \ldots \theta_p$ where $p = d - 2$ defined by

$$\begin{aligned} x_1 &= r\cos\theta_1, \qquad (4) \\ x_2 &= r\sin\theta_2\cos\theta_2, \\ x_3 &= r\sin\theta_1\sin\theta_2\cos\theta_3, \\ \ldots &= \ldots \\ x_p &= r\sin\theta_1\sin\theta_2, \ldots, \sin\theta_{p-1}, \cos\theta_p, \qquad (5) \\ x_{p+1} &= r\sin\theta_1\sin\theta_2, \ldots, \sin\theta_p, \cos\phi, \\ x_{p+2} &= r\sin\theta_1\sin\theta_2, \ldots, \sin\theta_p, \sin\phi, \end{aligned}$$

Again according to [1] the defining equations are

$$H(m_k, \pm, \vec{r}) = r^n Y(m_k, \theta_k, \pm\phi) \tag{6}$$

where

$$\begin{aligned} &Y(m_k, \theta_k, \pm\phi) \qquad (7) \\ &= \exp(\pm i m_p \phi) \prod_{k=0}^{p-1} (\sin\theta_{k+1})^{m_{k+1}} C_{m_k - m_{k-1}}^{m_{k+1}+1/2(p-k)}(\cos\theta_{k+1}). \end{aligned}$$

These $Y(m_k, \theta_k, \pm\phi)$ are complete and orthogonal on S^{d-1}. "In particular, any two surface harmonics of different degrees are orthognal on the unit-sphere". We will need the following formula for $N(m_0, m_1, \ldots, m_p)$ which will appear as the product of two spherical surface harmonics. We have

[1] Spherical surface harmonics are mostly called spherical harmonics for short.

$$E_k(l,m) \qquad (8)$$
$$= \frac{\pi 2^{k-2m-p}\Gamma(l+m+p+1-k)}{(l+1/2p+1/2-1/2k)(l-m)![\Gamma(m+1/2p+1/2-1/2k)]^2}$$

for any integers l, m where $l \geq m \geq 0$, and

$$N(m_0, m_1, \ldots, m_p) = 2\pi \prod_{k=1}^{n} E_k(m_{k-1}, m_k) \qquad (9)$$

where $m_0, m_1, \ldots, m_p$ satisfy (1).

Then with $\Omega = S^{d-1}$ the corresponding theorem is [1]

Any two distinct functions in (7) are orthogonal on Ω unless they are conjugate complex. In the case of conjugate complex functions we have

$$\int_\Omega \|Y(m_k, \theta, \pm\phi)\|^2 d\Omega = N(m_0, m_1, \ldots, m_p). \qquad (10)$$

A continuous function $f(\vec{x})$ on S^{d-1} can be expanded into the $Y(m_k, \theta_k, \pm\phi)$, refer to the literature for details. Further materials on harmonic polynomials and on spherical surface harmonics can be found in the literature [2, 3, 4]. Fabre gives a concise explanation of .....[3]. Dunkel and Yu give formula similar to (3) and (7) for complete sets of harmonic polynomials and spherical surface harmonics. Totik repeats one of these formulas [5]. However, the present author was not capable to implement these formulas into a program.

3 Explicit expressions for $Y(m_k, \theta_k, \pm\phi)$

This section provides the output from SSHY for dimension $d = 3, 4, 5$ and order $n = 1, \ldots, d$. Table 1 gives d and n in the first two columns. The third column contains the vector $m_0, \ldots, m_p$. The resulting function $Y(m_k, \theta_k, \pm\phi)$ is printed in the fourth column.

d	n	m	$Y(m_k, \theta_k, \pm\phi)$
3	1	[1 1]	$\exp(i * \phi) * \sin(\theta_1)$
3	1	[1 1]	$\exp(-i * \phi) * \sin(\theta_1)$
3	1	[1 0]	$\cos(\theta_1)$
3	2	[2 2]	$\exp(2 * i * \phi) * \sin(\theta_1)^2$
3	2	[2 2]	$\exp(-2 * i * \phi) * \sin(\theta_1)^2$
3	2	[2 1]	$3 * \exp(i * \phi) * \sin(\theta_1) * \cos(\theta_1)$
3	2	[2 1]	$3 * \exp(-i * \phi) * \sin(\theta_1) * \cos(\theta_1)$
3	2	[2 0]	$-1/2 + 3/2 * \cos(\theta_1)^2$

3	3	[3 3]	$\exp(3*i*\phi)*\sin(\theta_1)^3$
3	3	[3 3]	$\exp(-3*i*\phi)*\sin(\theta_1)^3$
3	3	[3 2]	$5*\exp(2*i*\phi)*\sin(\theta_1)^2*\cos(\theta_1)$
3	3	[3 2]	$5*\exp(-2*i*\phi)*\sin(\theta_1)^2*\cos(\theta_1)$
3	3	[3 1]	$3/2*\exp(i*\phi)*\sin(\theta_1)*(-1+5*\cos(\theta_1)^2)$
3	3	[3 1]	$3/2*\exp(-i*\phi)*\sin(\theta_1)*(-1+5*\cos(\theta_1)^2)$
3	3	[3 0]	$1/2*\cos(\theta_1)*(5*\cos(\theta_1)^2-3)$
4	1	[1 1 1]	$\exp(i*\phi)*\sin(\theta_2)*\sin(\theta_1)$
4	1	[1 1 1]	$\exp(-i*\phi)*\sin(\theta_2)*\sin(\theta_1)$
4	1	[1 1 0]	$\cos(\theta_2)*\sin(\theta_1)$
4	1	[1 0 0]	$2*\cos(\theta_1)$
4	2	[2 2 2]	$\exp(2*i*\phi)*\sin(\theta_2)^2*\sin(\theta_1)^2$
4	2	[2 2 2]	$\exp(-2*i*\phi)*\sin(\theta_2)^2*\sin(\theta_1)^2$
4	2	[2 2 1]	$3*\exp(i*\phi)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2$
4	2	[2 2 1]	$3*\exp(-i*\phi)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2$
4	2	[2 2 0]	$1/2*(-1+3*\cos(\theta_2)^2)*\sin(\theta_1)^2$
4	2	[2 1 1]	$4*\exp(i*\phi)*\sin(\theta_2)*\sin(\theta_1)*\cos(\theta_1)$
4	2	[2 1 1]	$4*\exp(-i*\phi)*\sin(\theta_2)*\sin(\theta_1)*\cos(\theta_1)$
4	2	[2 1 0]	$4*\cos(\theta_2)*\sin(\theta_1)*\cos(\theta_1)$
4	2	[2 0 0]	$-1+4*\cos(\theta_1)^2$
4	3	[3 3 3]	$\exp(3*i*\phi)*\sin(\theta_2)^3*\sin(\theta_1)^3$
4	3	[3 3 3]	$\exp(-3*i*\phi)*\sin(\theta_2)^3*\sin(\theta_1)^3$
4	3	[3 3 2]	$5*\exp(2*i*\phi)*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3$
4	3	[3 3 2]	$5*\exp(-2*i*\phi)*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3$
4	3	[3 3 1]	$3/2*\exp(i*\phi)*\sin(\theta_2)*(-1+5*\cos(\theta_2)^2)*\sin(\theta_1)^3$
4	3	[3 3 1]	$3/2*\exp(-i*\phi)*\sin(\theta_2)*(-1+5*\cos(\theta_2)^2)*\sin(\theta_1)^3$
4	3	[3 3 0]	$1/2*\cos(\theta_2)*(5*\cos(\theta_2)^2-3)*\sin(\theta_1)^3$
4	3	[3 2 2]	$6*\exp(2*i*\phi)*\sin(\theta_2)^2*\sin(\theta_1)^2*\cos(\theta_1)$
4	3	[3 2 2]	$6*\exp(-2*i*\phi)*\sin(\theta_2)^2*\sin(\theta_1)^2*\cos(\theta_1)$
4	3	[3 2 1]	$18*\exp(i*\phi)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2*\cos(\theta_1)$
4	3	[3 2 1]	$18*\exp(-i*\phi)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2*\cos(\theta_1)$
4	3	[3 2 0]	$3*(-1+3*\cos(\theta_2)^2)*\sin(\theta_1)^2*\cos(\theta_1)$
4	3	[3 1 1]	$2*\exp(i*\phi)*\sin(\theta_2)*\sin(\theta_1)*(-1+6*\cos(\theta_1)^2)$
4	3	[3 1 1]	$2*\exp(-i*\phi)*\sin(\theta_2)*\sin(\theta_1)*(-1+6*\cos(\theta_1)^2)$
4	3	[3 1 0]	$2*\cos(\theta_2)*\sin(\theta_1)*(-1+6*\cos(\theta_1)^2)$
4	3	[3 0 0]	$4*\cos(\theta_1)*(2*\cos(\theta_1)^2-1)$
4	4	[4 4 4]	$\exp(4*i*\phi)*\sin(\theta_2)^4*\sin(\theta_1)^4$
4	4	[4 4 4]	$\exp(-4*i*\phi)*\sin(\theta_2)^4*\sin(\theta_1)^4$
4	4	[4 4 3]	$7*\exp(3*i*\phi)*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4$
4	4	[4 4 3]	$7*\exp(-3*i*\phi)*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4$
4	4	[4 4 2]	$5/2*\exp(2*i*\phi)*\sin(\theta_2)^2*(-1+7*\cos(\theta_2)^2)*\sin(\theta_1)^4$
4	4	[4 4 2]	$5/2*\exp(-2*i*\phi)*\sin(\theta_2)^2*(-1+7*\cos(\theta_2)^2)*\sin(\theta_1)^4$

4	4	[4 4 1]	$5/2*\exp(i*\phi)*\sin(\theta_2)*\cos(\theta_2)*(7*\cos(\theta_2)^2-3)*\sin(\theta_1)^4$
4	4	[4 4 1]	$5/2*\exp(-i*\phi)*\sin(\theta_2)*\cos(\theta_2)*(7*\cos(\theta_2)^2-3)*\sin(\theta_1)^4$
4	4	[4 4 0]	$1/8*(3+35*\cos(\theta_2)^4-30*\cos(\theta_2)^2)*\sin(\theta_1)^4$
4	4	[4 3 3]	$8*\exp(3*i*\phi)*\sin(\theta_2)^3*\sin(\theta_1)^3*\cos(\theta_1)$
4	4	[4 3 3]	$8*\exp(-3*i*\phi)*\sin(\theta_2)^3*\sin(\theta_1)^3*\cos(\theta_1)$
4	4	[4 3 2]	$40*\exp(2*i*\phi)*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3*\cos(\theta_1)$
4	4	[4 3 2]	$40*\exp(-2*i*\phi)*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3*\cos(\theta_1)$
4	4	[4 3 1]	$12*\exp(i*\phi)*\sin(\theta_2)*(-1+5*\cos(\theta_2)^2)*\sin(\theta_1)^3*\cos(\theta_1)$
4	4	[4 3 1]	$12*\exp(-i*\phi)*\sin(\theta_2)*(-1+5*\cos(\theta_2)^2)*\sin(\theta_1)^3*\cos(\theta_1)$
4	4	[4 3 0]	$4*\cos(\theta_2)*(5*\cos(\theta_2)^2-3)*\sin(\theta_1)^3*\cos(\theta_1)$
4	4	[4 2 2]	$3*\exp(2*i*\phi)*\sin(\theta_2)^2*\sin(\theta_1)^2*(-1+8*\cos(\theta_1)^2)$
4	4	[4 2 2]	$3*\exp(-2*i*\phi)*\sin(\theta_2)^2*\sin(\theta_1)^2*(-1+8*\cos(\theta_1)^2)$
4	4	[4 2 1]	$9*\exp(i*\phi)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2*(-1+8*\cos(\theta_1)^2)$
4	4	[4 2 1]	$9*\exp(-i*\phi)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2*(-1+8*\cos(\theta_1)^2)$
4	4	[4 2 0]	$3/2*\sin(\theta_1)^2*(1-8*\cos(\theta_1)^2-3*\cos(\theta_2)^2+24*\cos(\theta_2)^2*\cos(\theta_1)^2)$
4	4	[4 1 1]	$4*\exp(i*\phi)*\sin(\theta_2)*\sin(\theta_1)*\cos(\theta_1)*(8*\cos(\theta_1)^2-3)$
4	4	[4 1 1]	$4*\exp(-i*\phi)*\sin(\theta_2)*\sin(\theta_1)*\cos(\theta_1)*(8*\cos(\theta_1)^2-3)$
4	4	[4 1 0]	$4*\cos(\theta_2)*\sin(\theta_1)*\cos(\theta_1)*(8*\cos(\theta_1)^2-3)$
4	4	[4 0 0]	$1+16*\cos(\theta_1)^4-12*\cos(\theta_1)^2$

5	1	[1 1 1 1]	$\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\sin(\theta_1)$
5	1	[1 1 1 1]	$\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\sin(\theta_1)$
5	1	[1 1 1 0]	$\cos(\theta_3)*\sin(\theta_2)*\sin(\theta_1)$
5	1	[1 1 0 0]	$2*\cos(\theta_2)*\sin(\theta_1)$
5	1	[1 0 0 0]	$3*\cos(\theta_1)$

5	2	[2 2 2 2]	$\exp(2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\sin(\theta_1)^2$
5	2	[2 2 2 2]	$\exp(-2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\sin(\theta_1)^2$
5	2	[2 2 2 1]	$3*\exp(i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\sin(\theta_1)^2$
5	2	[2 2 2 1]	$3*\exp(-i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\sin(\theta_1)^2$
5	2	[2 2 2 0]	$1/2*(-1+3*\cos(\theta_3)^2)*\sin(\theta_2)^2*\sin(\theta_1)^2$
5	2	[2 2 1 1]	$4*\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2$
5	2	[2 2 1 1]	$4*\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2$
5	2	[2 2 1 0]	$4*\cos(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2$
5	2	[2 2 0 0]	$(-1+4*\cos(\theta_2)^2)*\sin(\theta_1)^2$
5	2	[2 1 1 1]	$5*\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\sin(\theta_1)*\cos(\theta_1)$
5	2	[2 1 1 1]	$5*\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\sin(\theta_1)*\cos(\theta_1)$
5	2	[2 1 1 0]	$5*\cos(\theta_3)*\sin(\theta_2)*\sin(\theta_1)*\cos(\theta_1)$
5	2	[2 1 0 0]	$10*\cos(\theta_2)*\sin(\theta_1)*\cos(\theta_1)$
5	2	[2 0 0 0]	$-3/2+15/2*\cos(\theta_1)^2$

5	3	[3 3 3 3]	$\exp(3*i*\phi)*\sin(\theta_3)^3*\sin(\theta_2)^3*\sin(\theta_1)^3$
5	3	[3 3 3 3]	$\exp(-3*i*\phi)*\sin(\theta_3)^3*\sin(\theta_2)^3*\sin(\theta_1)^3$
5	3	[3 3 3 2]	$5*\exp(2*i*\phi)*\sin(\theta_3)^2*\cos(\theta_3)*\sin(\theta_2)^3*\sin(\theta_1)^3$
5	3	[3 3 3 2]	$5*\exp(-2*i*\phi)*\sin(\theta_3)^2*\cos(\theta_3)*\sin(\theta_2)^3*\sin(\theta_1)^3$
5	3	[3 3 3 1]	$3/2*\exp(i*\phi)*\sin(\theta_3)*(-1+5*\cos(\theta_3)^2)*\sin(\theta_2)^3*\sin(\theta_1)^3$

5	3	[3 3 3 1]	$3/2*\exp(-i*\phi)*\sin(\theta_3)*(-1+5*\cos(\theta_3)^2)*\sin(\theta_2)^3*\sin(\theta_1)^3$
5	3	[3 3 3 0]	$1/2*\cos(\theta_3)*(5*\cos(\theta_3)^2-3)*\sin(\theta_2)^3*\sin(\theta_1)^3$
5	3	[3 3 2 2]	$6*\exp(2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3$
5	3	[3 3 2 2]	$6*\exp(-2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3$
5	3	[3 3 2 1]	$18*\exp(i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3$
5	3	[3 3 2 1]	$18*\exp(-i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3$
5	3	[3 3 2 0]	$3*(-1+3*\cos(\theta_3)^2)*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3$
5	3	[3 3 1 1]	$2*\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)*(-1+6*\cos(\theta_2)^2)*\sin(\theta_1)^3$
5	3	[3 3 1 1]	$2*\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)*(-1+6*\cos(\theta_2)^2)*\sin(\theta_1)^3$
5	3	[3 3 1 0]	$2*\cos(\theta_3)*\sin(\theta_2)*(-1+6*\cos(\theta_2)^2)*\sin(\theta_1)^3$
5	3	[3 3 0 0]	$4*\cos(\theta_2)*(2*\cos(\theta_2)^2-1)*\sin(\theta_1)^3$
5	3	[3 2 2 2]	$7*\exp(2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\sin(\theta_1)^2*\cos(\theta_1)$
5	3	[3 2 2 2]	$7*\exp(-2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\sin(\theta_1)^2*\cos(\theta_1)$
5	3	[3 2 2 1]	$21*\exp(i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\sin(\theta_1)^2*\cos(\theta_1)$
5	3	[3 2 2 1]	$21*\exp(-i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\sin(\theta_1)^2*\cos(\theta_1)$
5	3	[3 2 2 0]	$7/2*(-1+3*\cos(\theta_3)^2)*\sin(\theta_2)^2*\sin(\theta_1)^2*\cos(\theta_1)$
5	3	[3 2 1 1]	$28*\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2*\cos(\theta_1)$
5	3	[3 2 1 1]	$28*\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2*\cos(\theta_1)$
5	3	[3 2 1 0]	$28*\cos(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2*\cos(\theta_1)$
5	3	[3 2 0 0]	$7*(-1+4*\cos(\theta_2)^2)*\sin(\theta_1)^2*\cos(\theta_1)$
5	3	[3 1 1 1]	$5/2*\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\sin(\theta_1)*(-1+7*\cos(\theta_1)^2)$
5	3	[3 1 1 1]	$5/2*\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\sin(\theta_1)*(-1+7*\cos(\theta_1)^2)$
5	3	[3 1 1 0]	$5/2*\cos(\theta_3)*\sin(\theta_2)*\sin(\theta_1)*(-1+7*\cos(\theta_1)^2)$
5	3	[3 1 0 0]	$5*\cos(\theta_2)*\sin(\theta_1)*(-1+7*\cos(\theta_1)^2)$
5	3	[3 0 0 0]	$5/2*\cos(\theta_1)*(7*\cos(\theta_1)^2-3)$

5	4	[4 4 4 4]	$\exp(4*i*\phi)*\sin(\theta_3)^4*\sin(\theta_2)^4*\sin(\theta_1)^4$
5	4	[4 4 4 4]	$\exp(-4*i*\phi)*\sin(\theta_3)^4*\sin(\theta_2)^4*\sin(\theta_1)^4$
5	4	[4 4 4 3]	$7*\exp(3*i*\phi)*\sin(\theta_3)^3*\cos(\theta_3)*\sin(\theta_2)^4*\sin(\theta_1)^4$
5	4	[4 4 4 3]	$7*\exp(-3*i*\phi)*\sin(\theta_3)^3*\cos(\theta_3)*\sin(\theta_2)^4*\sin(\theta_1)^4$
5	4	[4 4 4 2]	$5/2*\exp(2*i*\phi)*\sin(\theta_3)^2*(-1+7*\cos(\theta_3)^2)*\sin(\theta_2)^4*\sin(\theta_1)^4$
5	4	[4 4 4 2]	$5/2*\exp(-2*i*\phi)*\sin(\theta_3)^2*(-1+7*\cos(\theta_3)^2)*\sin(\theta_2)^4*\sin(\theta_1)^4$
5	4	[4 4 4 1]	$5/2*\exp(i*\phi)*\sin(\theta_3)*\cos(\theta_3)*(7*\cos(\theta_3)^2-3)*\sin(\theta_2)^4*\sin(\theta_1)^4$
5	4	[4 4 4 1]	$5/2*\exp(-i*\phi)*\sin(\theta_3)*\cos(\theta_3)*(7*\cos(\theta_3)^2-3)*\sin(\theta_2)^4*\sin(\theta_1)^4$
5	4	[4 4 4 0]	$1/8*(3+35*\cos(\theta_3)^4-30*\cos(\theta_3)^2)*\sin(\theta_2)^4*\sin(\theta_1)^4$
5	4	[4 4 3 3]	$8*\exp(3*i*\phi)*\sin(\theta_3)^3*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4$
5	4	[4 4 3 3]	$8*\exp(-3*i*\phi)*\sin(\theta_3)^3*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4$
5	4	[4 4 3 2]	$40*\exp(2*i*\phi)*\sin(\theta_3)^2*\cos(\theta_3)*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4$
5	4	[4 4 3 2]	$40*\exp(-2*i*\phi)*\sin(\theta_3)^2*\cos(\theta_3)*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4$
5	4	[4 4 3 1]	$12*\exp(i*\phi)*\sin(\theta_3)*(-1+5*\cos(\theta_3)^2)*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4$
5	4	[4 4 3 1]	$12*\exp(-i*\phi)*\sin(\theta_3)*(-1+5*\cos(\theta_3)^2)*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4$
5	4	[4 4 3 0]	$4*\cos(\theta_3)*(5*\cos(\theta_3)^2-3)*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4$
5	4	[4 4 2 2]	$3*\exp(2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*(-1+8*\cos(\theta_2)^2)*\sin(\theta_1)^4$
5	4	[4 4 2 2]	$3*\exp(-2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*(-1+8*\cos(\theta_2)^2)*\sin(\theta_1)^4$
5	4	[4 4 2 1]	$9*\exp(i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*(-1+8*\cos(\theta_2)^2)*\sin(\theta_1)^4$

5	4	[4 4 2 1]	$9*\exp(-i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*(-1+8*\cos(\theta_2)^2)*\sin(\theta_1)^4$
5	4	[4 4 2 0]	$3/2*\sin(\theta_2)^2*\sin(\theta_1)^4*(1-8*\cos(\theta_2)^2-3*\cos(\theta_3)^2+24*\cos(\theta_3)^2*\cos(\theta_2)^2)$
5	4	[4 4 1 1]	$4*\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*(8*\cos(\theta_2)^2-3)*\sin(\theta_1)^4$
5	4	[4 4 1 1]	$4*\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*(8*\cos(\theta_2)^2-3)*\sin(\theta_1)^4$
5	4	[4 4 1 0]	$4*\cos(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*(8*\cos(\theta_2)^2-3)*\sin(\theta_1)^4$
5	4	[4 4 0 0]	$(1+16*\cos(\theta_2)^4-12*\cos(\theta_2)^2)*\sin(\theta_1)^4$
5	4	[4 3 3 3]	$9*\exp(3*i*\phi)*\sin(\theta_3)^3*\sin(\theta_2)^3*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 3 3]	$9*\exp(-3*i*\phi)*\sin(\theta_3)^3*\sin(\theta_2)^3*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 3 2]	$45*\exp(2*i*\phi)*\sin(\theta_3)^2*\cos(\theta_3)*\sin(\theta_2)^3*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 3 2]	$45*\exp(-2*i*\phi)*\sin(\theta_3)^2*\cos(\theta_3)*\sin(\theta_2)^3*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 3 1]	$27/2*\exp(i*\phi)*\sin(\theta_3)*(-1+5*\cos(\theta_3)^2)*\sin(\theta_2)^3*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 3 1]	$27/2*\exp(-i*\phi)*\sin(\theta_3)*(-1+5*\cos(\theta_3)^2)*\sin(\theta_2)^3*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 3 0]	$9/2*\cos(\theta_3)*(5*\cos(\theta_3)^2-3)*\sin(\theta_2)^3*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 2 2]	$54*\exp(2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 2 2]	$54*\exp(-2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 2 1]	$162*\exp(i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 2 1]	$162*\exp(-i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 2 0]	$27*(-1+3*\cos(\theta_3)^2)*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 1 1]	$18*\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)*(-1+6*\cos(\theta_2)^2)*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 1 1]	$18*\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)*(-1+6*\cos(\theta_2)^2)*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 1 0]	$18*\cos(\theta_3)*\sin(\theta_2)*(-1+6*\cos(\theta_2)^2)*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 3 0 0]	$36*\cos(\theta_2)*(2*\cos(\theta_2)^2-1)*\sin(\theta_1)^3*\cos(\theta_1)$
5	4	[4 2 2 2]	$7/2*\exp(2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\sin(\theta_1)^2*(-1+9*\cos(\theta_1)^2)$
5	4	[4 2 2 2]	$7/2*\exp(-2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\sin(\theta_1)^2*(-1+9*\cos(\theta_1)^2)$
5	4	[4 2 2 1]	$21/2*\exp(i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\sin(\theta_1)^2*(-1+9*\cos(\theta_1)^2)$
5	4	[4 2 2 1]	$21/2*\exp(-i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\sin(\theta_1)^2*(-1+9*\cos(\theta_1)^2)$
5	4	[4 2 2 0]	$7/4*\sin(\theta_2)^2*\sin(\theta_1)^2*(1-9*\cos(\theta_1)^2-3*\cos(\theta_3)^2+27*\cos(\theta_3)^2*\cos(\theta_1)^2)$
5	4	[4 2 1 1]	$14*\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2*(-1+9*\cos(\theta_1)^2)$
5	4	[4 2 1 1]	$14*\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2*(-1+9*\cos(\theta_1)^2)$
5	4	[4 2 1 0]	$14*\cos(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2*(-1+9*\cos(\theta_1)^2)$
5	4	[4 2 0 0]	$7/2*\sin(\theta_1)^2*(1-9*\cos(\theta_1)^2-4*\cos(\theta_2)^2+36*\cos(\theta_2)^2*\cos(\theta_1)^2)$
5	4	[4 1 1 1]	$35/2*\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\sin(\theta_1)*\cos(\theta_1)*(3*\cos(\theta_1)^2-1)$
5	4	[4 1 1 1]	$35/2*\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\sin(\theta_1)*\cos(\theta_1)*(3*\cos(\theta_1)^2-1)$
5	4	[4 1 1 0]	$35/2*\cos(\theta_3)*\sin(\theta_2)*\sin(\theta_1)*\cos(\theta_1)*(3*\cos(\theta_1)^2-1)$
5	4	[4 1 0 0]	$35*\cos(\theta_2)*\sin(\theta_1)*\cos(\theta_1)*(3*\cos(\theta_1)^2-1)$
5	4	[4 0 0 0]	$15/8+315/8*\cos(\theta_1)^4-105/4*\cos(\theta_1)^2$

5	5	[5 5 5 5]	$\exp(5*i*\phi)*\sin(\theta_3)^5*\sin(\theta_2)^5*\sin(\theta_1)^5$
5	5	[5 5 5 5]	$\exp(-5*i*\phi)*\sin(\theta_3)^5*\sin(\theta_2)^5*\sin(\theta_1)^5$
5	5	[5 5 5 4]	$9*\exp(4*i*\phi)*\sin(\theta_3)^4*\cos(\theta_3)*\sin(\theta_2)^5*\sin(\theta_1)^5$

5	5	[5 5 5 4]	$9*\exp(-4*i*\phi)*\sin(\theta_3)^4*\cos(\theta_3)*\sin(\theta_2)^5*\sin(\theta_1)^5$
5	5	[5 5 5 3]	$7/2*\exp(3*i*\phi)*\sin(\theta_3)^3*(-1+9*\cos(\theta_3)^2)*\sin(\theta_2)^5*\sin(\theta_1)^5$
5	5	[5 5 5 3]	$7/2*\exp(-3*i*\phi)*\sin(\theta_3)^3*(-1+9*\cos(\theta_3)^2)*\sin(\theta_2)^5*\sin(\theta_1)^5$
5	5	[5 5 5 2]	$35/2*\exp(2*i*\phi)*\sin(\theta_3)^2*\cos(\theta_3)*(-1+3*\cos(\theta_3)^2)*\sin(\theta_2)^5*\sin(\theta_1)^5$
5	5	[5 5 5 2]	$35/2*\exp(-2*i*\phi)*\sin(\theta_3)^2*\cos(\theta_3)*(-1+3*\cos(\theta_3)^2)*\sin(\theta_2)^5*\sin(\theta_1)^5$
5	5	[5 5 5 1]	$15/8*\exp(i*\phi)*\sin(\theta_3)*(1+21*\cos(\theta_3)^4-14*\cos(\theta_3)^2)*\sin(\theta_2)^5*\sin(\theta_1)^5$
5	5	[5 5 5 1]	$15/8*\exp(-i*\phi)*\sin(\theta_3)*(1+21*\cos(\theta_3)^4-14*\cos(\theta_3)^2)*\sin(\theta_2)^5*\sin(\theta_1)^5$
5	5	[5 5 5 0]	$1/8*\cos(\theta_3)*(63*\cos(\theta_3)^4-70*\cos(\theta_3)^2+15)*\sin(\theta_2)^5*\sin(\theta_1)^5$
5	5	[5 5 4 4]	$10*\exp(4*i*\phi)*\sin(\theta_3)^4*\sin(\theta_2)^4*\cos(\theta_2)*\sin(\theta_1)^5$
5	5	[5 5 4 4]	$10*\exp(-4*i*\phi)*\sin(\theta_3)^4*\sin(\theta_2)^4*\cos(\theta_2)*\sin(\theta_1)^5$
5	5	[5 5 4 3]	$70*\exp(3*i*\phi)*\sin(\theta_3)^3*\cos(\theta_3)*\sin(\theta_2)^4*\cos(\theta_2)*\sin(\theta_1)^5$
5	5	[5 5 4 3]	$70*\exp(-3*i*\phi)*\sin(\theta_3)^3*\cos(\theta_3)*\sin(\theta_2)^4*\cos(\theta_2)*\sin(\theta_1)^5$
5	5	[5 5 4 2]	$25*\exp(2*i*\phi)*\sin(\theta_3)^2*(-1+7*\cos(\theta_3)^2)*\sin(\theta_2)^4*\cos(\theta_2)*\sin(\theta_1)^5$
5	5	[5 5 4 2]	$25*\exp(-2*i*\phi)*\sin(\theta_3)^2*(-1+7*\cos(\theta_3)^2)*\sin(\theta_2)^4*\cos(\theta_2)*\sin(\theta_1)^5$
5	5	[5 5 4 1]	$25*\exp(i*\phi)*\sin(\theta_3)*\cos(\theta_3)*(7*\cos(\theta_3)^2-3)*\sin(\theta_2)^4*\cos(\theta_2)*\sin(\theta_1)^5$
5	5	[5 5 4 1]	$25*\exp(-i*\phi)*\sin(\theta_3)*\cos(\theta_3)*(7*\cos(\theta_3)^2-3)*\sin(\theta_2)^4*\cos(\theta_2)*\sin(\theta_1)^5$
5	5	[5 5 4 0]	$5/4*(3+35*\cos(\theta_3)^4-30*\cos(\theta_3)^2)*\sin(\theta_2)^4*\cos(\theta_2)*\sin(\theta_1)^5$
5	5	[5 5 3 3]	$4*\exp(3*i*\phi)*\sin(\theta_3)^3*\sin(\theta_2)^3*(-1+10*\cos(\theta_2)^2)*\sin(\theta_1)^5$
5	5	[5 5 3 3]	$4*\exp(-3*i*\phi)*\sin(\theta_3)^3*\sin(\theta_2)^3*(-1+10*\cos(\theta_2)^2)*\sin(\theta_1)^5$
5	5	[5 5 3 2]	$20*\exp(2*i*\phi)*\sin(\theta_3)^2*\cos(\theta_3)*\sin(\theta_2)^3*(-1+10*\cos(\theta_2)^2)*\sin(\theta_1)^5$
5	5	[5 5 3 2]	$20*\exp(-2*i*\phi)*\sin(\theta_3)^2*\cos(\theta_3)*\sin(\theta_2)^3*(-1+10*\cos(\theta_2)^2)*\sin(\theta_1)^5$
5	5	[5 5 3 1]	$6*\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)^3*\sin(\theta_1)^5*(1-10*\cos(\theta_2)^2-5*\cos(\theta_3)^2+50*\cos(\theta_3)^2*\cos(\theta_2)^2)$
5	5	[5 5 3 1]	$6*\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)^3*\sin(\theta_1)^5*(1-10*\cos(\theta_2)^2-5*\cos(\theta_3)^2+50*\cos(\theta_3)^2*\cos(\theta_2)^2)$
5	5	[5 5 3 0]	$2*\cos(\theta_3)*\sin(\theta_2)^3*\sin(\theta_1)^5*(-5*\cos(\theta_3)^2+50*\cos(\theta_3)^2*\cos(\theta_2)^2+3-30*\cos(\theta_2)^2)$
5	5	[5 5 2 2]	$8*\exp(2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\cos(\theta_2)*(10*\cos(\theta_2)^2-3)*\sin(\theta_1)^5$
5	5	[5 5 2 2]	$8*\exp(-2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\cos(\theta_2)*(10*\cos(\theta_2)^2-3)*\sin(\theta_1)^5$
5	5	[5 5 2 1]	$24*\exp(i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\cos(\theta_2)*(10*\cos(\theta_2)^2-3)*\sin(\theta_1)^5$
5	5	[5 5 2 1]	$24*\exp(-i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\cos(\theta_2)*(10*\cos(\theta_2)^2-3)*\sin(\theta_1)^5$
5	5	[5 5 2 0]	$4*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^5*(-10*\cos(\theta_2)^2+3+30*\cos(\theta_3)^2*\cos(\theta_2)^2-9*\cos(\theta_3)^2)$

5	5	[5 5 1 1]	$\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)*(3+80*\cos(\theta_2)^4-48*\cos(\theta_2)^2)*\sin(\theta_1)^5$
5	5	[5 5 1 1]	$\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)*(3+80*\cos(\theta_2)^4-48*\cos(\theta_2)^2)*\sin(\theta_1)^5$
5	5	[5 5 1 0]	$\cos(\theta_3)*\sin(\theta_2)*(3+80*\cos(\theta_2)^4-48*\cos(\theta_2)^2)*\sin(\theta_1)^5$
5	5	[5 5 0 0]	$2*\cos(\theta_2)*(16*\cos(\theta_2)^4-16*\cos(\theta_2)^2+3)*\sin(\theta_1)^5$
5	5	[5 4 4 4]	$11*\exp(4*i*\phi)*\sin(\theta_3)^4*\sin(\theta_2)^4*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 4 4]	$11*\exp(-4*i*\phi)*\sin(\theta_3)^4*\sin(\theta_2)^4*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 4 3]	$77*\exp(3*i*\phi)*\sin(\theta_3)^3*\cos(\theta_3)*\sin(\theta_2)^4*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 4 3]	$77*\exp(-3*i*\phi)*\sin(\theta_3)^3*\cos(\theta_3)*\sin(\theta_2)^4*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 4 2]	$55/2*\exp(2*i*\phi)*\sin(\theta_3)^2*(-1+7*\cos(\theta_3)^2)*\sin(\theta_2)^4*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 4 2]	$55/2*\exp(-2*i*\phi)*\sin(\theta_3)^2*(-1+7*\cos(\theta_3)^2)*\sin(\theta_2)^4*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 4 1]	$55/2*\exp(i*\phi)*\sin(\theta_3)*\cos(\theta_3)*(7*\cos(\theta_3)^2-3)*\sin(\theta_2)^4*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 4 1]	$55/2*\exp(-i*\phi)*\sin(\theta_3)*\cos(\theta_3)*(7*\cos(\theta_3)^2-3)*\sin(\theta_2)^4*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 4 0]	$11/8*(3+35*\cos(\theta_3)^4-30*\cos(\theta_3)^2)*\sin(\theta_2)^4*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 3 3]	$88*\exp(3*i*\phi)*\sin(\theta_3)^3*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 3 3]	$88*\exp(-3*i*\phi)*\sin(\theta_3)^3*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 3 2]	$440*\exp(2*i*\phi)*\sin(\theta_3)^2*\cos(\theta_3)*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 3 2]	$440*\exp(-2*i*\phi)*\sin(\theta_3)^2*\cos(\theta_3)*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 3 1]	$132*\exp(i*\phi)*\sin(\theta_3)*(-1+5*\cos(\theta_3)^2)*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 3 1]	$132*\exp(-i*\phi)*\sin(\theta_3)*(-1+5*\cos(\theta_3)^2)*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 3 0]	$44*\cos(\theta_3)*(5*\cos(\theta_3)^2-3)*\sin(\theta_2)^3*\cos(\theta_2)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 2 2]	$33*\exp(2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*(-1+8*\cos(\theta_2)^2)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 2 2]	$33*\exp(-2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*(-1+8*\cos(\theta_2)^2)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 2 1]	$99*\exp(i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*(-1+8*\cos(\theta_2)^2)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 2 1]	$99*\exp(-i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*(-1+8*\cos(\theta_2)^2)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 2 0]	$33/2*\sin(\theta_2)^2*\sin(\theta_1)^4*\cos(\theta_1)*(1-8*\cos(\theta_2)^2-3*\cos(\theta_3)^2+24*\cos(\theta_3)^2*\cos(\theta_2)^2)$
5	5	[5 4 1 1]	$44*\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*(8*\cos(\theta_2)^2-3)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 1 1]	$44*\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*(8*\cos(\theta_2)^2-3)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 1 0]	$44*\cos(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*(8*\cos(\theta_2)^2-3)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 4 0 0]	$11*(1+16*\cos(\theta_2)^4-12*\cos(\theta_2)^2)*\sin(\theta_1)^4*\cos(\theta_1)$
5	5	[5 3 3 3]	$9/2*\exp(3*i*\phi)*\sin(\theta_3)^3*\sin(\theta_2)^3*\sin(\theta_1)^3*(-1+11*\cos(\theta_1)^2)$

5	5	[5 3 3 3]	$9/2*\exp(-3*i*\phi)*\sin(\theta_3)^3*\sin(\theta_2)^3*\sin(\theta_1)^3*(-1+11*\cos(\theta_1)^2)$
5	5	[5 3 3 2]	$45/2*\exp(2*i*\phi)*\sin(\theta_3)^2*\cos(\theta_3)*\sin(\theta_2)^3*\sin(\theta_1)^3*(-1+11*\cos(\theta_1)^2)$
5	5	[5 3 3 2]	$45/2*\exp(-2*i*\phi)*\sin(\theta_3)^2*\cos(\theta_3)*\sin(\theta_2)^3*\sin(\theta_1)^3*(-1+11*\cos(\theta_1)^2)$
5	5	[5 3 3 1]	$27/4*\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)^3*\sin(\theta_1)^3*(1-11*\cos(\theta_1)^2-5*\cos(\theta_3)^2+55*\cos(\theta_3)^2*\cos(\theta_1)^2)$
5	5	[5 3 3 1]	$27/4*\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)^3*\sin(\theta_1)^3*(1-11*\cos(\theta_1)^2-5*\cos(\theta_3)^2+55*\cos(\theta_3)^2*\cos(\theta_1)^2)$
5	5	[5 3 3 0]	$9/4*\cos(\theta_3)*\sin(\theta_2)^3*\sin(\theta_1)^3*(-5*\cos(\theta_3)^2+55*\cos(\theta_3)^2*\cos(\theta_1)^2+3-33*\cos(\theta_1)^2)$
5	5	[5 3 2 2]	$27*\exp(2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3*(-1+11*\cos(\theta_1)^2)$
5	5	[5 3 2 2]	$27*\exp(-2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3*(-1+11*\cos(\theta_1)^2)$
5	5	[5 3 2 1]	$81*\exp(i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3*(-1+11*\cos(\theta_1)^2)$
5	5	[5 3 2 1]	$81*\exp(-i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3*(-1+11*\cos(\theta_1)^2)$
5	5	[5 3 2 0]	$27/2*\sin(\theta_2)^2*\cos(\theta_2)*\sin(\theta_1)^3*(1-11*\cos(\theta_1)^2-3*\cos(\theta_3)^2+33*\cos(\theta_3)^2*\cos(\theta_1)^2)$
5	5	[5 3 1 1]	$9*\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\sin(\theta_1)^3*(1-11*\cos(\theta_1)^2-6*\cos(\theta_2)^2+66*\cos(\theta_2)^2*\cos(\theta_1)^2)$
5	5	[5 3 1 1]	$9*\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\sin(\theta_1)^3*(1-11*\cos(\theta_1)^2-6*\cos(\theta_2)^2+66*\cos(\theta_2)^2*\cos(\theta_1)^2)$
5	5	[5 3 1 0]	$9*\cos(\theta_3)*\sin(\theta_2)*\sin(\theta_1)^3*(1-11*\cos(\theta_1)^2-6*\cos(\theta_2)^2+66*\cos(\theta_2)^2*\cos(\theta_1)^2)$
5	5	[5 3 0 0]	$18*\cos(\theta_2)*\sin(\theta_1)^3*(-2*\cos(\theta_2)^2+22*\cos(\theta_2)^2*\cos(\theta_1)^2+1-11*\cos(\theta_1)^2)$
5	5	[5 2 2 2]	$21/2*\exp(2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\sin(\theta_1)^2*\cos(\theta_1)*(11*\cos(\theta_1)^2-3)$
5	5	[5 2 2 2]	$21/2*\exp(-2*i*\phi)*\sin(\theta_3)^2*\sin(\theta_2)^2*\sin(\theta_1)^2*\cos(\theta_1)*(11*\cos(\theta_1)^2-3)$
5	5	[5 2 2 1]	$63/2*\exp(i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\sin(\theta_1)^2*\cos(\theta_1)*(11*\cos(\theta_1)^2-3)$
5	5	[5 2 2 1]	$63/2*\exp(-i*\phi)*\sin(\theta_3)*\cos(\theta_3)*\sin(\theta_2)^2*\sin(\theta_1)^2*\cos(\theta_1)*(11*\cos(\theta_1)^2-3)$
5	5	[5 2 2 0]	$21/4*\sin(\theta_2)^2*\sin(\theta_1)^2*\cos(\theta_1)*(-11*\cos(\theta_1)^2+3+33*\cos(\theta_3)^2*\cos(\theta_1)^2-9*\cos(\theta_3)^2)$
5	5	[5 2 1 1]	$42*\exp(i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2*\cos(\theta_1)*(11*\cos(\theta_1)^2-3)$
5	5	[5 2 1 1]	$42*\exp(-i*\phi)*\sin(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2*\cos(\theta_1)*(11*\cos(\theta_1)^2-3)$
5	5	[5 2 1 0]	$42*\cos(\theta_3)*\sin(\theta_2)*\cos(\theta_2)*\sin(\theta_1)^2*\cos(\theta_1)*(11*\cos(\theta_1)^2-3)$
5	5	[5 2 0 0]	$21/2*\sin(\theta_1)^2*\cos(\theta_1)*(-11*\cos(\theta_1)^2+3+44*\cos(\theta_2)^2*\cos(\theta_1)^2-12*\cos(\theta_2)^2)$

5	5	[5 1 1 1]	$35/8 * \exp(i * \phi) * \sin(\theta_3) * \sin(\theta_2) * \sin(\theta_1) * (1 + 33 * \cos(\theta_1)^4 - 18 * \cos(\theta_1)^2)$
5	5	[5 1 1 1]	$35/8 * \exp(-i * \phi) * \sin(\theta_3) * \sin(\theta_2) * \sin(\theta_1) * (1 + 33 * \cos(\theta_1)^4 - 18 * \cos(\theta_1)^2)$
5	5	[5 1 1 0]	$35/8 * \cos(\theta_3) * \sin(\theta_2) * \sin(\theta_1) * (1 + 33 * \cos(\theta_1)^4 - 18 * \cos(\theta_1)^2)$
5	5	[5 1 0 0]	$35/4 * \cos(\theta_2) * \sin(\theta_1) * (1 + 33 * \cos(\theta_1)^4 - 18 * \cos(\theta_1)^2)$
5	5	[5 0 0 0]	$21/8 * \cos(\theta_1) * (33 * \cos(\theta_1)^4 - 30 * \cos(\theta_1)^2 + 5)$

4 Usage of SSHY

This section describes the usage of SSHY. Essentially section 4 is a LaTeXexport of a Maple worksheet. We save us the work to format the export neatly, Maple definitely needs a better export routine to LaTeX. Within this worksheet three examples are given for a call to SSHY. Furthermore, the orthogonality of the $Y(m_k, \theta_k, \pm\phi)$ is tested. The integral should evaluate to zero for orthognal functions. Finally, the integral over the product of two conjugate complex spherical harmonics is formed explicitly and the resulting number is compared with formula (9). The resulting two numbers should be equal. We hope that the worksheet is self-explanatory.

d-dimensional Spherical Surface Harmonics ($d >= 3$)

Thomas Wieder

thomas.wieder@t-online.de

July 2009

1 Introduction

The program SSHY calculates the Spherical Surface Harmonics (SSH) Y(d,n,theta,phi) according to the definition by Harry Bateman.

d = dimension, n = order, theta, phi = hyperspherical polar coordinates (with radius r =1).

2 Initialization

```
> restart;
```

These lines show how to call SSHY with symbolic arguments. Spherical coordinates are used.

$$d := 3$$
$$n := 2$$
$$\phi := \phi$$
$$\theta := [\ 0\ ]$$
$$\theta_1 := theta1$$
$$x := [\ 0 \quad 0 \quad 0\ ]$$
$$coordsys := \text{“}spherical''$$

$$\text{“}''$$
$$\text{“}SSHY : d, n, m, theta[1], theta[p], phi :'',\ 3,\ 2,\ theta1,\ theta1,\ \phi$$
$$\text{“}SSHY : Numberoffunctionstobecalculated :'',\ 5$$
$$\text{“}''$$
$$3,\ 2,\ \text{“}+'',\ e^{2 \cdot i \cdot \phi} \cdot (\sin(theta1))^2$$
$$3,\ 2,\ \text{“}-'',\ e^{`-`(2 \cdot i \cdot \phi)} \cdot (\sin(theta1))^2$$
$$3,\ 2,\ \text{“}+'',\ 3 \cdot e^{i \cdot \phi} \cdot \sin(theta1) \cdot \cos(theta1)$$
$$3,\ 2,\ \text{“}-'',\ 3 \cdot e^{`-`(i \cdot \phi)} \cdot \sin(theta1) \cdot \cos(theta1)$$
$$3,\ 2,\ \text{“}+'',\ `-`(1 \cdot 1/2) + 3 \cdot 1 \cdot 1/2 \cdot (\cos(theta1))^2$$

These lines show how to call SSHY with numeric arguments. Spherical coordinates are used.

$$d := 3$$
$$n := 2$$
$$\phi := \pi$$
$$\theta := [\ 0\ ]$$
$$\theta_1 := 1 \cdot 1/2 \cdot \pi$$
$$x := [\ 0 \quad 0 \quad 0\ ]$$
$$coordsys := \text{“}spherical''$$

$$\text{“}''$$
$$\text{“}SSHY : d, n, m, theta[1], theta[p], phi :'',\ 3,\ 2,\ 1 \cdot 1/2 \cdot \pi,\ 1 \cdot 1/2 \cdot \pi,\ \pi$$
$$\text{“}SSHY : Numberoffunctionstobecalculated :'',\ 5$$
$$\text{“}''$$
$$3,\ 2,\ \text{“}+'',\ 1$$
$$3,\ 2,\ \text{“}-'',\ 1$$
$$3,\ 2,\ \text{“}+'',\ 0$$
$$3,\ 2,\ \text{“}-'',\ 0$$

$$3,\ 2,\ \text{“}+\text{''},\ \text{‘-‘}\ (1\cdot 1/2)$$

These lines show how to call SSHY with symbolic arguments. Cartesian coordinates are used, therefore the resulting functions are not surface spherical harmonics (SSH) but spherical Harmonics (SH). That is SH = r^n SHY.

$$d := 3$$
$$n := 2$$
$$\phi := \phi$$
$$\theta := \left[\ 0\ \right]$$
$$x := \left[\ 0 \quad 0 \quad 0\ \right]$$
$$x_1 := x1$$
$$x_2 := x2$$
$$x_3 := x3$$
$$coordsys := \text{“}cartesian\text{''}$$

$$\text{“''}$$
$$\text{“}SSHY : d, n, m, x(1), x[d] :\text{''},\ 3,\ 2,\ x1,\ x3$$
$$\text{“}SSHY : Numberoffunctionstobecalculated :\text{''},\ 5$$
$$\text{“''}$$
$$3,\ 2,\ \text{“}+\text{''},\ (x2 + i\cdot x3)^2$$
$$3,\ 2,\ \text{“}-\text{''},\ \left(x2^2 + x3^2\right)^2 \cdot 1 \cdot (x2 + i\cdot x3)^{-2}$$
$$3,\ 2,\ \text{“}+\text{''},\ 3\,(x2 + i\cdot x3)\cdot x1$$
$$3,\ 2,\ \text{“}-\text{''},\ 3\left(x2^2 + x3^2\right)\cdot x1 \cdot 1 \cdot (x2 + i\cdot x3)^{-1}$$
$$3,\ 2,\ \text{“}+\text{''},\ x1^2 + \text{‘-‘}\left(1\cdot 1/2\cdot x2^2\right) + \text{‘-‘}\left(1\cdot 1/2\cdot x3^2\right)$$

Transfrom the SSH from cartesian coordintate to spherical coordinates. Under the assumption that r = 1, show the identity of a SSH expressed in both coordinate systems.

Example with d := 3 and n := 2.

$$SSH := 3\,(x2 + i\cdot x3)\cdot x1$$
$$SSH := 3\,(x2 + i\cdot x3)\cdot \cos(theta1)$$
$$SSH := 3\,(\sin(theta1)\cdot\cos(\phi) + i\cdot x3)\cdot\cos(theta1)$$
$$SSH := 3\,(\sin(theta1)\cdot\cos(\phi) + i\cdot\sin(theta1)\cdot\sin(\phi))\cdot\cos(theta1)$$
$$SSH := 3\cdot\cos(theta1)\cdot\sin(theta1)\cdot e^{i\cdot\phi}$$

```
> with(VectorCalculus):
```

Prove that any two different SSH are orthogonal. Use SSH expressed in spherical coordinates. The surface integral must be equal to 0.

Example with d := 3 and n := 2.

$$f := e^{3 \cdot i \cdot \phi} \cdot (\sin(\mathit{theta1}))^3$$

$$g := 5 \cdot e^{`-`(2 \cdot i \cdot \phi)} \cdot (\sin(\mathit{theta1}))^2 \cdot \cos(\mathit{theta1})$$

$$0$$

Prove that the product of any SSH with its compex conjugate equals formula Formula 11.3 (5). Example with d := 3 and n := 2.

$$f := 5 \cdot e^{2 \cdot i \cdot \phi} \cdot (\sin(\mathit{theta1}))^2 \cdot \cos(\mathit{theta1})$$

$$g := 5 \cdot e^{`-`(2 \cdot i \cdot \phi)} \cdot (\sin(\mathit{theta1}))^2 \cdot \cos(\mathit{theta1})$$

$$160 \cdot 1 \cdot 1/21 \cdot r^2 \cdot \pi$$

$$23.93594403 \cdot r^2$$

$$2.425218182 \cdot \pi^2$$

$$23.93594405$$

3 Conclusions

We have shown how to calcultate Spherical Surface Hamronics (SSH) of dimension d and order n.

4 References

Harry Bateman (Staff Of The Bateman Manuscript Project, Editors), Higher Transcendental Functions, Vol. II,

McGraw-Hill Book Company, New York, 1953, Chapters 11.2 Harmonic Functions and 11.3 Surface Harmonics, Formula 11.2 (23), page 240.

6 SSHY.mpl

This section contains a listing of the source code of SSHY.mpl which was written in the Maple 12 programming language. SSHY implements the formulas (3), (7), and (9). The listing is longer than the very implementation of the formulas would require. Several precautions were implemented to reject invalid input. Furthermore, some optional print statements were included for debugging purposes.

```
SSHY :=
proc(d::integer,n::integer,theta::Array,phi,x::Array,coordsys::string)

# Start: 19 June 2009.
# Last change: 23 July 2009.
# thomas.wieder@t-online.de

# Calculate a complete set of orthogonal surface spherical harmonics
# (SSH) at point (r,theta,phi).
# The SSH are also called hyperspherical harmonics for d > 3.
# See subroutine SSHY for literature on SSH.
# Written with Maple 12.

# ----------------------------------------------------------------------
# These lines show how to call SSHY with symbolic arguments.
# Here we are using spherical coordinates theta[1],...,theta[p], phi,
# then x is a dummy argument.
# The resulting functions are surface spherical harmonics SSH in
# theta1,theta2,...,phi.
# ----------------------------------------------------------------------
# > d := 3;
# > n := 2;
# > phi := 'phi';
# > theta := Array(1 .. d-2);
```

```
# > for k to d-2 do theta[k] := cat('theta', k) end do;
# > x := Array(1 .. d);
# > coordsys := "spherical";
#
# > SSHY(d, n, theta, phi, x, coordsys);
#
#                                   " "
# "SSHY: d, n, m, theta[1], theta[p], phi:", 3, 2, theta1, theta1, phi
#              "SSHY: Number of functions to be calculated:", 5
#                                   " "
#                                                               2
#           "SSHY: Y(d,n,theta,phi) = ", exp(2 I phi) sin(theta1)
#
#                                                                2
#           "SSHY: Y(d,n,theta,phi) = ", exp(-2 I phi) sin(theta1)
#
#      "SSHY: Y(d,n,theta,phi) = ", 3 exp(I phi) sin(theta1) cos(theta1)
#
#     "SSHY: Y(d,n,theta,phi) = ", 3 exp(-I phi) sin(theta1) cos(theta1)
#
#                                          1   3           2
#              "SSHY: Y(d,n,theta,phi) = ", - - + - cos(theta1)
#                                          2   2

# ----------------------------------------------------------------------
# These lines show how to call SSHY with numeric arguments.
# Here we are using spherical coordinates theta[1],...,theta[p], phi,
# then x is a dummy argument.
# ----------------------------------------------------------------------
# > d := 3;
# > n := 2;
# > phi := Pi;
# > theta := Array(1 .. d-2);
# > for k to d-2 do theta[k] := Pi/(k+1) end do;
# > x := Array(1 .. d);
# > coordsys := "spherical";
#
# > SSHY(d, n, theta, phi, x, coordsys);
#
#                                                      1     1
#    "SSHY: d, n, m, theta[1], theta[p], phi:", 3, 2, - Pi, - Pi,Pi
#                                                      2     2
#              "SSHY: Number of functions to be calculated:", 5
#                                   " "
#                       "SSHY: Y(d,n,theta,phi) = ", 1
#                       "SSHY: Y(d,n,theta,phi) = ", 1
```

```
#                        "SSHY: Y(d,n,theta,phi) = ", 0
#                        "SSHY: Y(d,n,theta,phi) = ", 0
#                                                     -1
#                        "SSHY: Y(d,n,theta,phi) = ", --
#                                                     2

# ----------------------------------------------------------------------
# These lines show how to call SSHY with symbolic arguments.
# Here we are using cartesian coordinates x[1],...,x[d],
# then theta[1],...,theta[p] and phi are dummy arguments.
# The resulting functions are spherical harmonics SH in x1,x2,...,xd.
# ----------------------------------------------------------------------
# > d := 3;
# > n := 2;
# > phi := 'phi';
# > theta := Array(1 ..d-2);
# > x := Array(1 .. d);
# > for k to d do x[k] := cat('x', k) end do;
# > coordsys := "cartesian";
#
# > SSHY(d, n, theta, phi, x, coordsys);
#
#                                  " "
#               "SSHY: d, n, m, x(1), x[d]:", 3, 2, 0, 0, x1, x3
#               "SSHY: Number of functions to be calculated:", 5
#                                  " "
#                                                           2
#                   "SSHY: Y(d,n,theta,phi) = ", (x2 + I x3)
#
#                                                           2
#                                              /  2     2\
#                                              \x2  + x3 /
#                   "SSHY: Y(d,n,theta,phi) = ", -----------
#                                                           2
#                                              (x2 + I x3)
#
#                 "SSHY: Y(d,n,theta,phi) = ", 3 (x2 + I x3) x1
#
#                                              /  2     2\
#                                            3 \x2  + x3 / x1
#                 "SSHY: Y(d,n,theta,phi) = ", ---------------
#                                               x2 + I x3
#
#                                            2   1   2   1   2
#               "SSHY: Y(d,n,theta,phi) = ", x1  - - x2  - - x3
#                                                2       2
```

```
local iverbose, dmin, dmax, nmin, nmax, m, p, k, tpl, ny, Ydn,
NumberOfFunctionsY,  Result;
global Mvec, NumberOfTupels;

description
"Calculate Spherical Surface Harmonics Y according to Harry Bateman's formula.";

Result := Array(1..2);

# ----------------------------------------------------------------------
# Set up some global parameters.
# ----------------------------------------------------------------------

# Parameter iverbose sets the debugging level.
# No debugging output for iverbose = 0.
iverbose := 0;

# Minimal and maximal order:
nmin := 0;
nmax := 8;
# Minmial and maximal dimension:
dmin := 3;
dmax := 8;

# ----------------------------------------------------------------------
# Check input data.
# ----------------------------------------------------------------------

try

if n < nmin or n > nmax then
print("SSHY: Wrong order n! Choose 1 <= n <= nmax ! n =",n);
error "Error in subroutine SSHY!";
return;
end if;

if d < dmin or d > dmax then
print("SSHY: Wrong dimension d! Choose 3 <= d <= dmax ! d =",d);
error "Error in subroutine SSHY!";
return;
end if;

if coordsys <> "spherical" and coordsys <> "cartesian" then
print("SSHY: Wrong coordinate system specified! Choose spherical or cartesian !");
error "Error in subroutine SSHY!";
```

```
return;
end if;

if coordsys = "spherical" then

if ArrayNumElems(theta) < d-2 then
print("SSHY: Wrong dimensions for array theta!");
print("ArrayNumElems(theta)=",ArrayNumElems(theta));
end if;

for k from 1 to d-2 do
if type(theta[k], algnum) then
if theta[k] < 0 or theta[k] > 2*Pi then
print("SSHY: Wrong angle theta[k]!");
print("Choose 0 >= theta[k] >= 2 Pi ! k, theta[k]:",k,theta[k]);
error "Error in subroutine SSHY!";
return;
end if;
end if;
end do;

if type(phi, algnum) then
if phi < 0.0 or phi > 2*Pi then
print("SSHY: Wrong angle phi! Choose 0 >= phi >= 2 Pi ! phi =",phi);
error "Error in subroutine SSHY!";
return;
end if;
end if;

elif coordsys = "cartesian" then

if ArrayNumElems(x) < d then
print("SSHY: Wrong dimensions for array x! ArrayNumElems(x)=",ArrayNumElems(x));
end if;

# End if-clause coordsys.
end if;

catch:
printf("SSHY: Something went wrong: %q\n",lastexception);
error
end try;

# ----------------------------------------------------------------------
# Set up some parameters.
# ----------------------------------------------------------------------
```

```
p := d - 2;
m := Array(0..p);
NumberOfFunctionsY := (2*n+p)*(n+p-1)!/(p!*n!);
Ydn := Array(1..NumberOfFunctionsY);

# ---------------------------------------------------------------------
# Echo some input data to the user.
# ---------------------------------------------------------------------

print(" ");
if coordsys = "spherical" then
print("SSHY: d, n, m, theta[1], theta[p], phi:",d,n,theta[1],theta[p],phi);
elif coordsys = "cartesian" then
print("SSHY: d, n, m, x(1), x[d]:",d,n,x[1],x[d]);
end if;
print("SSHY: Number of functions to be calculated:",NumberOfFunctionsY);
print(" ");

# ---------------------------------------------------------------------
# Populate the matrix Mvec.
# ---------------------------------------------------------------------

# The matrix Mvec has NumberOfTupels rows and p+1 columns.
# Each row of Mvec contains a tupel of integers which generates a
# corresponding SSH.

dnp(d,n);

if iverbose = 1 then
print("SSHY: Populate the matrix Mvec:");
print("SSHY: d, n, NumberOfTupels:",d,n,NumberOfTupels);
printf("%d\n",Mvec);
end if;

# ---------------------------------------------------------------------
# Calculate a complete orthogonal set of the surface spherical
# harmonics.
# ---------------------------------------------------------------------

ny := 0;
for tpl from 1 to NumberOfTupels do

# The vector m corresponds to one row of matrix Mvec.
# The vector m has p+1 elements m[0],...,m[p].
# This is a possible vector m for d=5, n=3, and p=d-2=3:
```

```
# 3 3 3 3
# 3 3 3 2
# 3 3 3 1
# 3 3 3 0
# 3 3 2 2
# 3 3 2 1
# 3 3 2 0
# 3 3 1 1
# 3 3 1 0
# 3 3 0 0
# 3 2 2 2
# 3 2 2 1
# 3 2 2 0
# 3 2 1 1
# 3 2 1 0
# 3 2 0 0
# 3 1 1 1
# 3 1 1 0
# 3 1 0 0
# 3 0 0 0
# See subroutine dnp(d,n).

for k from 0 to p do
m[k] := Mvec[tpl,k];
end do;

# -----------------------------------------------------------------------
# Calculate Ydn for the given tupel m.
# -----------------------------------------------------------------------

if iverbose = 1 then
print("SSHY: Will call SSHY_Formula...coordsys =",coordsys);
end if;
if coordsys = "spherical" then
Result := SSHY_Formula_Spherical(d,n,m,theta,phi);
elif coordsys = "cartesian" then
Result := SSHY_Formula_Cartesian(d,n,m,x);
end if;

# Result contains 2 functions because of the argument +-phi or +-m[p].
ny := ny + 1;
Ydn[ny] := Result[1];
#printf("%d %c %d %c %c %d %c %c %a
#%s\n",d,"&",n,"&","[",m,"]","&",Ydn[ny],"\\\\");
#printf("%d %d %c %d %c %a\n",d,n,"[",m,"]",Ydn[ny]);
#printf("%d %d %d a%\n",d,n,m,Ydn[ny]);
```

```
print(d,n,"+",Ydn[ny]);
if m[p] > 0 then
ny := ny + 1;
Ydn[ny] := Result[2];
#printf("%d %c %d %c %c %d %c %c %a
#%s\n",d,"&",n,"&","[",m,"]","&",Ydn[ny],"\\\\");
#printf("%d %d %c %d %c %a\n",d,n,"[",m,"]",Ydn[ny]);
#printf("%d %d %d a%\n",d,n,m,Ydn[ny]);
print(d,n,"-",Ydn[ny]);
end if;

end do;

# End of procedure SSHY.
end proc;




SSHY_Formula_Spherical :=
proc(d::integer,n::integer,m::Array,theta::Array,phi)

# Start: 19 June 2009.
# Last change: 5 July 2009.
# thomas.wieder@t-online.de

# Calculate the surface spherical harmonics Ynd for the given parameters
# d, n, m
# at the given point (r,theta,phi) on the surface of the n-dimensional
# hypersphere.
# The spherical harmonics are expressed in spherical coordinates
# theta[1],..,theta[p], phi.

# Literature:
# Harry Bateman (Staff Of The Bateman Manuscript Project, Editors),
# Higher Transcendental Functions, Vol. II,
# McGraw-Hill Book Company, New York, 1953,
# Chapters 11.2 Harmonic Functions and 11.3 Surface Harmonics,
# Formula 11.2 (23), page 240.

local iverbose, p, Ydn, k, factor1, factor2, Result;
Result := Array(1..2);

# ---------------------------------------------------------------------
# Set up some parameters.
# ---------------------------------------------------------------------
```

```
# Parameter iverbose sets the debugging level.
# No debugging output for iverbose = 0.
iverbose := 0;

p := d - 2;

if iverbose=1 then
print("SSHY_Formula_Spherical: d, n, p:",d,n,p);
print("SSHY_Formula_Spherical: Number of elements of array m",ArrayNumElems(m));
print("SSHY_Formula_Spherical: m[0], m[p]:",m[0],m[p]);
printf("%d\n",m);
end if;

# ---------------------------------------------------------------------
# Check input data.
# ---------------------------------------------------------------------

try

for k from 1 to p do
if iverbose = 1 then
print("SSHY_Formula_Spherical: k,m[k],theta[k]",k,m[k],theta[k]);
end if;
if m[k] > m[k-1] then
print("SSHY_Formula_Spherical: Wrong m[k]!");
print("Choose n = m[0] >= m[1] >=... m[p] >= 0! k, m[k]:",k,m[k]);
error "Error in subroutine SSHY_Formula_Spherical!";
return;
end if;
end do;

catch:
printf("SSHY_Formula_Spherical: Something went wrong: %q\n",lastexception);
error
end try;

# ---------------------------------------------------------------------
# Calculate a single surface spherical harmonics Ynd for the given
# arguments.
# ---------------------------------------------------------------------

Ydn := 1;

for k from 0 to p-1 do
```

```
factor1 := sin(theta[k+1])^m[k+1];
factor2 := GegenbauerC(m[k]-m[k+1],m[k+1]+(1/2)*(p-k),cos(theta[k+1]));
Ydn := factor1 * factor2 * Ydn;

if iverbose = 1 then
print("SSHY_Formula_Spherical: factor1, factor2, Ydn:",factor1,factor2,Ydn);
end if;

# end of do-loop *** k ***.
end do;

Result[1] := exp(+I*m[p]*phi)*Ydn;
Result[2] := exp(-I*m[p]*phi)*Ydn;
Result[1] := simplify(Result[1]);
Result[2] := simplify(Result[2]);

Result;

# End of procedure SSHY_Formula_Spherical.
end proc;
```

```
SSHY_Formula_Cartesian := proc(d::integer,n::integer,m::Array,x::Array)

# Start: 19 June 2009.
# Last change: 9 July 2009.
# thomas.wieder@t-online.de

# Calculate the surface spherical harmonics Ynd for the given parameters
# d, n, m
# at the given point (r,theta,phi) on the surface of the n-dimensional
# hypersphere.
# The spherical harmonics are expressed in cartesian coordinates
# x[1],..,x[p].

# Literature:
# Harry Bateman (Staff Of The Bateman Manuscript Project, Editors),
# Higher Transcendental Functions, Vol. II,
# McGraw-Hill Book Company, New York, 1953,
# Chapters 11.2 Harmonic Functions and 11.3 Surface Harmonics,
# Formula 11.2 (21), page 239.

local iverbose, p, Ydn, k, kk, factor1, factor2, rk, rp, Result;
```

```
Result := Array(1..2);

# ----------------------------------------------------------------------
# Set up some parameters.
# ----------------------------------------------------------------------

# Parameter iverbose sets the debugging level.
# No debugging output for iverbose = 0.
iverbose := 0;

p := d - 2;

if iverbose=1 then
print("SSHY_Formula_Cartesian: d, n, p:",d,n,p);
print("SSHY_Formula_Cartesian: Number of elements of array m",ArrayNumElems(m));
print("SSHY_Formula_Cartesian: m[0], m[p]:",m[0],m[p]);
printf("%d\n",m);
end if;

# ----------------------------------------------------------------------
# Check input data.
# ----------------------------------------------------------------------

try

for k from 1 to p do
if iverbose = 1 then
print("SSHY_Formula_Cartesian: k,m[k],x[k]",k,m[k],x[k]);
end if;
if m[k] > m[k-1] then
print("SSHY_Formula_Cartesian: Wrong m[k]!");
print("Choose n = m[0] >= m[1] >=... m[p] >= 0! k, m[k]:",k,m[k]);
error "Error in subroutine SSHY_Formula_Cartesian!";
return;
end if;
end do;

catch:
printf("SSHY_Formula_Cartesian: Something went wrong:%q\n",lastexception);
error
end try;

# ----------------------------------------------------------------------
# Calculate a single surface spherical harmonics Ynd for the given
arguments.
# ----------------------------------------------------------------------
```

```
rp := sqrt(x[p+1]^2+x[p+2]^2);
if rp = 0.0 then
print("SSHY_Formula_Cartesian: r[p] = 0! p+1, x[p+1], p+2, x[p+2]:");
print(p+1,x[p+1],p+2,x[p+2]);
error "Error in subroutine SSHY_Formula_Cartesian!";
return;
end if;

Ydn := 1;

for k from 0 to p-1 do

rk := 0.0;
for kk from k+1 to p+2 do
rk := rk + x[kk]^2;
end do;
rk := sqrt(rk);

factor1 := rk^(m[k]-m[k+1]);
factor2 := GegenbauerC(m[k]-m[k+1],m[k+1]+(1/2)*(p-k),x[k+1]/rk);
#factor2 := GegenbauerC(m[k+1]+(1/2)*(p-k),m[k]-m[k+1],x[k+1]/rk);
Ydn := factor1 * factor2 * Ydn;

if iverbose = 1 then
print("SSHY_Formula_Cartesian: factor1, factor2, Ydn:",factor1,factor2,Ydn)
end if;

# end of do-loop *** k ***.
end do;

Result[1] := ((x[p+1]/rp+I*x[p+2]/rp)^m[p])*(rp^m[p])*Ydn;
Result[2] := ((x[p+1]/rp+I*x[p+2]/rp)^(-m[p]))*(rp^m[p])*Ydn;
Result[1] := simplify(Result[1]);
Result[2] := simplify(Result[2]);

Result;

# End of procedure SSHY_Formula_Cartesian.
end proc;




test_dnp := proc()
```

```
local d, n, p, tpl, k;
global Tupel,Mvec;
for d from 5 to 5 do
p := d - 2;
for n from 1 to 5 do
dnp(d,n);
print("test_dnp: d, n, Result =");
print("d,n:",d,n);
printf("%4d\n",Mvec);
end do;
end do;
# End of procedure test_dnp.
end proc;
```

```
dnp := proc(d::integer, n::integer)

# Begonnen am: 22.6.2009
# Letzte nderung am: 28.6.2009

local iverbose, p, k, tpl;
global Mvec, Tupel, NumberOfTupels;

# This is the matrix Mvec for d=5 and n=3.
# 3 3 3 3
# 3 3 3 2
# 3 3 3 1
# 3 3 3 0
# 3 3 2 2
# 3 3 2 1
# 3 3 2 0
# 3 3 1 1
# 3 3 1 0
# 3 3 0 0
# 3 2 2 2
# 3 2 2 1
# 3 2 2 0
# 3 2 1 1
# 3 2 1 0
# 3 2 0 0
# 3 1 1 1
# 3 1 1 0
# 3 1 0 0
# 3 0 0 0
```

```
# ---------------------------------------------------------------------
# Check input data.
# ---------------------------------------------------------------------

if d < 1 then
print("dnp: Wrong dimension d! Choose d > 1! d =",d);
error "Error in subroutine dnp!";
return;
end if;

if n < 0 then
print("dnp: Wrong order n! Choose n > 0! n =",n);
error "Error in subroutine dnp!";
return;
end if;

# ---------------------------------------------------------------------
# Set up some parameters.
# ---------------------------------------------------------------------

# Parameter iverbose sets the debugging level.
# No debugging output for iverbose = 0.
iverbose := 0;

p := d - 2;

NumberOfTupels := binomial(n+p,p);
if iverbose = 1 then
print("dnp: NumberOfTupels =",NumberOfTupels);
end if;

Mvec := Array(1..NumberOfTupels,0...p);
Tupel := Array(0..p);

# ---------------------------------------------------------------------
# Populate matrix Mvec.
# ---------------------------------------------------------------------

tpl := 1;
for k from 0 to p do
Mvec[tpl,k] := n
end do;

while Mvec[tpl,1] > 0 do
```

```
for k from 0 to p do
Tupel[k] := Mvec[tpl,k];
end do;

dnp2(n, p);

tpl := tpl + 1;
for k from 0 to p do
Mvec[tpl,k] := Tupel[k];
end do;

if iverbose = 1 then
print(" ");
print("dnp: tpl =",tpl);
printf("%4d\n",Mvec);
end if;

end do;

# End of procedure dnp.
end proc;




dnp2 := proc(n::integer, p::integer)

# Begonnen am: 22.6.2009
# Letzte nderung am: 23.6.2009

local iverbose, i, j;
global Mvec, Tupel;

iverbose := 0;

for i from p to 0 by -1 do

if Tupel[i] > 0 then
Tupel[i] := Tupel[i] - 1;

for j from p to i by -1 do
Tupel[j] := Tupel[i];
end do;

if iverbose = 1 then
print(" ");
```

```
printf("%4d\n",Tupel);
end if;

return Tupel;
end if;

end do;

# End of procedure dnp2.
end proc;




NmOmp := proc(p::integer, m::Array)

# Begonnen am: 4.7.2009
# Letzte nderung am: 4.7.2009

# Literature:
# Harry Bateman (Staff Of The Bateman Manuscript Project, Editors)
# Higher Transcendental Functions, Vol. II,
# McGraw-Hill Book Company, New York, 1953
# Chapters 11.2 Harmonic Functions and 11.3 Surface Harmonics
# Formula 11.3 (5), page 240.

#> f := 5*exp((2*I)*phi)*sin(theta1)^2*cos(theta1);
#                                             2
#                   5 exp(2 I phi) sin(theta1)  cos(theta1)
#
#> g := 5*exp(-(2*I)*phi)*sin(theta1)^2*cos(theta1);
#                                             2
#                  5 exp(-2 I phi) sin(theta1)  cos(theta1)
#> `assuming`([SurfaceInt(f*g, [r, theta1, phi] = Surface(`<,>`(r,
# theta, phi), theta = 0 .. Pi, phi = 0 .. 2*Pi, coords = spherical))], [r = 1]);
#                                  160  2
#                                  --- r  Pi
#                                  21
#> evalf((160/21)*Pi);
#                                 23.93594403
#> mTest := Array(0 .. 1); mTest[0] := 3; mTest[1] := 2; mTest;
#                            Array(%id = 76890728)
#                                      3
#                                      2
#                            Array(%id = 76890728)
#> NmOmp(1, mTest);
```

```
#                                            2
#                                2.425218182 Pi
#> evalf(%);
#                                  23.93594405
local k,Result;
Result := 1;
for k from 1 to p do
Result := Result*Eklm(k,m[k-1],m[k],p);
end do;
Result := Result*2*Pi;
# End of procedure NmOmp.
end proc;
```

```
Eklm := proc(k::integer,l::integer,m::integer,p::integer)
local Result;
Result := Pi*2^(k-2*m-p)*GAMMA(l+m+p+1-k);
Result := Result/(l+0.5*p+0.5-0.5*k);
Result := Result/(l-m)!;
Result := Result/GAMMA(m+0.5*p+0.5-0.5*k)^2;
# End of procedure Eklm.
end proc;
```

7 Conlusion

SSHY gives complete sets of harmonic polynomials or complete sets of spherical (surface) harmonics for (in principle) arbitrary dimension d and order n. However, a sufficently complex program can always be improved. Any corresponding adive will be welcomed as well as any hints on errors or bugs.

Acknowledgements Many thanks to Charles F. Dunkel and Yuan Xu for fruitful hints on definitions and literature.

References

[1] Harry Bateman (Staff Of The Bateman Manuscript Project, Editors), Higher Transcendental Functions, Vol. II, McGraw-Hill Book Company, New York, 1953, Chapters 11.2 Harmonic Functions and 11.3 Surface Harmonics, Formula 11.2 (23), page 240.

[2] Claus Müller, Spherical Harmonics, Berlin, 1966, Springer Verlag.

[3] M. Fabre de la Ripell, The Hyperspherical Expansion Method, in: Models and methods in few-body physics, L.S. Ferreira (ed.), Springer Verlag, Berlin, 1987. pp. 283 - 323.

[4] Charles F. Dunkel and Yuan Xu, Orthogonal Polynomials of Several Variables, Cambridge University Press, Cambridge, 2001, Chapter 2.2.

[5] Vilmos Totik, Orthogonal polynomials, in: Surveys in Approximation Theory, vol. 1, 2005, pp. 70-125.