SHBundle 2021β

Nico Sneeuw, Matthias Weigelt, Markus Antoni,
Matthias Roth, Balaji Devaraju et al.

affiliation: Institute of Geodesy, University of Stuttgart

October 22, 2021

Terms of use

In publications, we suggest the following acknowledgment:

The toolbox SHBUNDLE *(version #/date #) was developed by N. Sneeuw, M. Weigelt, B. Devaraju, M. Roth, M. Antoni et. al. and its latest version is provided via download from the Institute of Geodesy (GIS), University of Stuttgart:*
`http://www.gis.uni-stuttgart.de/research/projects/Bundles/`.

For feedback, questions or bugs reports, please contact bundle@gis.uni-stuttgart.de

1 Content and history of the SHBUNDLE

The program package SHBUNDLE provides basic tools for

- spherical harmonic synthesis
- spherical harmonic analysis

and the routines are implemented in MATLAB and OCATVE.

According to the documentation, the first routines were created by Nico Sneeuw since 1994 at the IAPG, TU München, and collected until 2000 to form the 1st version of the SHBUNDLE . The software was updated 2002 and 2008 by Nico Sneeuw and Matthias Weigelt and others to versions 2 and 3, respectively. Version 4 contains the modifications and additional tools, written by Matthias Weigelt at the Institute of Geodesy of the University of Stuttgart (GIS) until 2012.

In 2013, the SHBUNDLE V.4 was modified and a few tools have been added to form SHBUNDLE 2013. Tools, which are not related to spherical harmonics, are removed from this package and will become part of other software bundles.

From this version onwards, we decided to make the software available to the public. Please feel free to make comments on the software, notify us about bugs or even provide useful enhancements via bundle@gis.uni-stuttgart.de.

Please consider:

- For complete functionality, the package UBERALL from the same source is necessary.
- The routine *sph_gradshs.m* cannot be executed in OCATVE, because function handles on internal routines are not implemented there.
- Some users reported, that the function *legendreP.m* is shadowed by a MATLAB-built-in routine with the same name but different behavior. Unfortunately, this built-in routine has a higher priority also after re-ordering of the search path entries. Removing the path of the SYMBOLIC TOOLBOX enables to our knowledge a proper use of the SHBUNDLE 2021β.

See the changelog for the feature changes in the latest version.

1.1 Changelog

SHBUNDLE 2021 β:

- new function *evp_gruenbaum.m* determines the spherical harmonic coefficients of the Slepian basis functions (within a spherical cap) via the eigenvalue problem of Grünbaum.
- new function *getSlepianSHC.m* rearranges the Grünbaum eigenvectors into `|C\S|`-format per Slepian basis function and applies a rotation of the center to another location on the sphere.
- new feature in *gshs_grid.m* enables spherical harmonic synthesis for multiple sets of spherical harmonic coefficients – given in a 3D matrix – e.g. monthly solutions of a year or a set of Slepian basis functions. (A demonstration can be found in the routine *example_slepian.m*.)
- included Slepian basis function into documentation

SHBUNDLE 2021 α:

- new functions *wigner_all.m* and *rotate_shc.m* enable the rotation of spherical harmonic functions or coefficients via the Wigner-d-functions.
- *ylmplot.m* can also visualize the derivatives of the spherical harmonic functions now.
- deprecated *gcoef2sc.m* was removed and replaced by *clm2sc.m*.
- deprecated *gshs.m* was removed and replaced by *gshs_.m*.
- deprecated *gshsag.m* was removed and replaced by*gshs_grid.m*.
- deprecated *gshscovfn.m* and *gshscovfnag.m* were removed and replaced by *gshs2ptfun.m*.
- deprecated *gshsptw.m* was removed and replaced by *gshs_ptw.m*.
- deprecated *plmspeed.m* was removed and replaced by *Legendre_mex.m*.
- deprecated *rshs.m* was removed and replaced by *regionalshs.m*.
- updated documentation

SHBUNDLE 2015 α:

- change in *plm.m*: output has now always length of ϑ rows and length of l columns; explanation: *plm.m* produced a vector in the same orientation of either ϑ or l if one of them was a vector and the other one a scalar! this behavior made additional transposing operations necessary which made programming more complex and, additionally, inflicted a small calculation speed loss,
- fork *gshsptw.m* to *gshs_ptw.m*, use getopt-style parameters and implement speed-up (for best speed: feed data chunks of size 500 to functions like *plm.m* or *Legendre_mex.m*); declare *gshsptw.m* deprecated,
- fork *gshs.m* to *gshs_.m*, use getopt-style parameters, declare *gshs.m* deprecated,
- fork *gshsag.m* to *gshs_grid.m*, use getopt-style parameters, declare *gshs_grid.m* deprecated,
- new function *gshs2ptfun.m* replaces *gshscovfn.m* and *gshscovfnag.m* (both declared deprecated),
- new function *regionalshs.m* replaces *rshs.m* (which is declared deprecated),
- rewrite C-code of *Legendre_mex.m*: split to several files for better overview,
- updated documentation.

SHBUNDLE 2014:

- change to radians is finished, remove the somewhat unpractical "_rad" from filenames,
- behaviour change for consistency: *plm.m* now calculates derivatives towards co-latitude (not latitude anymore) – *check your code!*
- *legendreP.m* was replaced by the faster *legendreP.m* (suitable for $N_{\max} < 1500$), *legendreP_Xnum.m* (suitable for $N_{\max} \geq 1500$) and *legendreIndex.m*,
- *Legendre_mex.m* (kept) and *Legendre_x_mex.m* (removed) were merged, the behaviour of *plm.m* was added to form a new enhanced version of *Legendre_mex.m*, however, *plm.m* will be kept as well for persons without C-compiler),
- *gcoef2sc.m* (deprecated) and *icgem2sc.m* (removed) were replaced/enhanced by *clm2sc.m*,
- *icgem_parser.m* was replaced/enhanced by *parse_icgem.m*,
- updated documentation,
- all functions were reviewed to unify the help-text.

SHBUNDLE 2013: In this version, the input and output of angles (longitude, latitude, co-latitude) has been changed from degree to radian for consistency. (The suffix "RAD" was added to the angles in this process for a better control. Please feel free to remove it.) All functions which act on spherical coordinates get the suffix "_rad".

If you have worked with an older version of the SHBUNDLE 2013, check in particular the usage of

- *gshs.m*
- *gshsag.m*
- *gshsptw.m*
- *intpnm.m*
- *iplm.m*
- *iplmquad.m*
- *legpol.m*
- *normg.m*
- *plm.m*
- *rshs.m*
- *ylm.m*

in your code.

The optional smoothing of the field by *pellinen.m*, *gaussian.m* or *destriping.m* was removed from the routines

- *deginfo.m*
- *gshsag.m*
- *gshs.m*
- *rshs.m*
- *gshsptw.m*
- *isotf.m*/*eigengrav.m*

together with input argument 'cap'. If you want to apply any filter on the spherical harmonic coefficients, please use the (upcoming) filter toolbox.

- The functions *isvec.m* and *isscal.m* have been replaced by the MATLAB-internal routines (*isvector.m* and *isscalar.m*).
- The function *leggrad.m* was replaced by the better documented routine *legendreP.m* with additional features.
- The function *isotf.m* has been replaced by *eigengrav.m*, which neglects the filter option.

2 Spherical harmonics

According to potential theory, the gravitational field of each body is a harmonic function and fulfills the Laplace equation. Due to spherical geometry of the Earth (or other celestial bodies) the Laplace equation is written in spherical coordinates

$$r^2\Delta\Phi = r^2\frac{\partial^2\Phi}{\partial r^2} + 2r\frac{\partial\Phi}{\partial r} + \frac{\partial^2\Phi}{\partial\vartheta^2} + \cot\vartheta\frac{\partial\Phi}{\partial\vartheta} + \frac{1}{\sin^2\vartheta}\frac{\partial^2\Phi}{\partial\lambda} = 0 \tag{2.1}$$

where λ denotes the longitude, ϑ the co-latitude and r the radius.

The ansatz of separation $\Phi(\lambda,\vartheta,r) = g(\vartheta)h(\lambda)f(r)$ leads to 3 ordinary differential equations, one for each coordinate

$$r^2\frac{\mathrm{d}^2 f}{\mathrm{d}r^2} + 2r\frac{\mathrm{d}f}{\mathrm{d}r} - n(n+1)f = 0, \tag{2.2}$$

$$\frac{\mathrm{d}^2 h}{\mathrm{d}\lambda^2} + m^2 h = 0, \tag{2.3}$$

$$\frac{\mathrm{d}^2 g}{\mathrm{d}\vartheta^2} + \frac{\mathrm{d}g}{\mathrm{d}\vartheta}\cot\vartheta + \left(n(n+1) - \frac{m^2}{\sin^2\vartheta}\right) g = 0, \tag{2.4}$$

where the m is referred to as order and n as the degree. Please keep in mind, that many authors use the letter l instead of n for the degree.

The solution of equation (2.2) and (2.3) are the functions $f(r) \in \{r^n, r^{-(n+1)}\}$, respectively $h(\lambda) \in \{\cos m\lambda, \sin m\lambda\}$. The Legendre differential equation (2.4) is solved by the associated Legendre functions[1] of first and second kind $g(\vartheta) \in \{P_{n,m}(\cos\vartheta), Q_{n,m}(\cos\vartheta)\}$.

Any linear combination of f, g, h

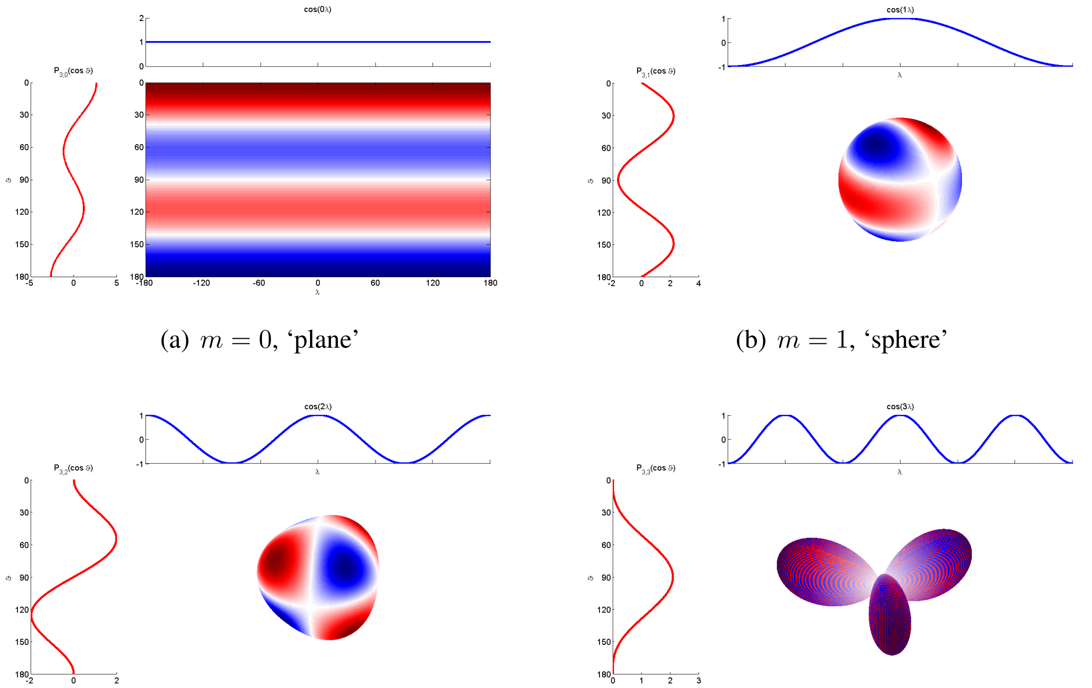
$$\Phi(\lambda,\vartheta,r) = \sum_{n=0}^{\infty}\sum_{m=0}^{n} \alpha_{n,m} \begin{Bmatrix} P_{n,m}(\cos\vartheta) \\ Q_{n,m}(\cos\vartheta) \end{Bmatrix} \cdot \begin{Bmatrix} \cos m\lambda \\ \sin m\lambda \end{Bmatrix} \cdot \begin{Bmatrix} r^n \\ r^{-(n+1)} \end{Bmatrix}, \tag{2.5}$$

with arbitrary coefficients $\alpha_{n,m} \in \mathbb{R}$, is a solution of the Laplace equation.

The gravitational potential should

- tend to zero for infinite distances ($r \to \infty$)
- be limited on the sphere

[1]The notation of degree n and order m is often without separation $P_{nm}(\cos\vartheta)$ or as upper and lower index $P_n^m(\cos\vartheta)$, but we try to be consistent in this documentation.



(a) $m = 0$, 'plane' (b) $m = 1$, 'sphere'

(c) $m = 2$, 'geoid' (d) $m = 3$, 'orbital'

Figure 2.1: Visualization of spherical harmonics of degree $n = 3$ with the different options of *ylmplot.m*.

and so the functions $Q_{n,m}(\cos\vartheta)$ and r^n are excluded from the model. For numerical reasons, the normalized[2] Legendre functions

$$\overline{P}_{n,m}(\cos\vartheta) = P_{n,m}(\cos\vartheta)\sqrt{(2-\delta_{m0})(2n+1)\frac{(n-m)!}{(n+m)!}} \tag{2.6}$$

are used in the following.

Up-scaling the unit sphere to the radius R and normalization of the Legendre functions lead to the gravitational potential V in the outer space $r \geq R$ of the Earth:

$$V(\lambda,\vartheta,r) = \frac{GM}{R}\sum_{n=0}^{N_{\max}}\sum_{m=0}^{n}\left(\frac{R}{r}\right)^{n+1}\overline{P}_{n,m}(\cos\vartheta)\Big(\overline{C}_{n,m}\cos m\lambda + \overline{S}_{n,m}\sin m\lambda\Big) \tag{2.7}$$

The summation over coefficients $\{\overline{C}_{n,m}, \overline{S}_{n,m}\}$ and spherical basis functions is denoted as **Spherical Harmonic Synthesis** (SHS), while further factors – like the gravitational constant G or the mass of the Earth M – determine the field quantity.

[2]The standard in our package is the geodetic normalization (2.6), which is preferred in geodetic textbooks and assumed to global gravity models in terms of spherical harmonic coefficients of CSR, GFZ, JPL, … For the rotation of spherical harmonic coefficients and functions, alternative 'complex' normalizations are introduced (see section 3.1, and the routines *LeNorm.m*, *cpx*.m* …)

Compact Notation: In publications, often a short notation of the spherical harmonics synthesis is introduced by

$$V(\lambda, \vartheta, r) = \frac{GM}{R} \sum_{n=0}^{N_{\max}} \left(\frac{R}{r}\right)^{n+1} \sum_{m=-n}^{n} \overline{K}_{n,m} \overline{Y}_{n,m}(\lambda, \vartheta). \tag{2.8}$$

This implies the convention

$$\begin{aligned} \overline{K}_{n,m} &= \begin{cases} \overline{C}_{n,m} & \text{for } m \geq 0 \\ \overline{S}_{n,|m|} & \text{for } m < 0 \end{cases} \\ \overline{Y}_{n,m} &= \begin{cases} \overline{P}_{n,m}(\cos\vartheta)\cos(m\lambda) & \text{for } m \geq 0 \\ \overline{P}_{n,|m|}(\cos\vartheta)\sin(|m|\lambda) & \text{for } m < 0, \end{cases} \end{aligned} \tag{2.9}$$

where the modulus of the order m is often neglected in the notation.

2.1 Legendre functions

The associated **Legendre functions** (in the geodetic normalization) are given by

$$\overline{P}_{n,m}(t) := \sqrt{(2-\delta_{m0})(2n+1)\frac{(n-m)!}{(n+m)!}} \sqrt{1-t^2}^{\,m} \frac{\mathrm{d}^{n+m}}{\mathrm{d}t^{n+m}} \frac{1}{2^n n!}(t^2-1)^n \tag{2.10}$$

with degree n, order m and the convenient substitution $t = \cos\vartheta$. The Legendre functions are calculated by the recursion[3]

$$\overline{P}_{0,0}(t) = 1 \tag{2.11a}$$

$$\overline{P}_{m,m}(t) = W_{m,m} \sin\vartheta\, \overline{P}_{m-1,m-1}(t) \qquad \text{for } m > 0 \text{ and } m = n \tag{2.11b}$$

$$\overline{P}_{n,m}(t) = W_{n,m}\left[\cos\vartheta\, \overline{P}_{n-1,m}(t) - \frac{1}{W_{n-1,m}} \overline{P}_{n-2,m}(t)\right] \qquad \text{for } m \neq n \tag{2.11c}$$

with the (coefficient) factors

$$W_{n,m} = \begin{cases} \sqrt{3} & \text{for } n = 1 \text{ and } m = \{0, 1\} \\ \sqrt{\frac{2n+1}{2n}} & \text{if } n = m \text{ and } n > 1 \\ \sqrt{\frac{(2n+1)(2n-1)}{(n+m)(n-m)}} & n > 1 \text{ and } m \neq n \end{cases} \tag{2.12}$$

and the convention $\overline{P}_{n,m}(t) = 0$ for $m > n$. This well-known algorithm is implemented in *legendreP.m* and it is shown to be stable until degree $n \approx 1800$ by Xue and Sneeuw (2006). For higher degrees and order, the stabilized (but slightly slower) version with X-numbers (Fukushima, 2012), i. e. *legendreP_Xnum.m* is recommended.

Additionally, a speed-optimized mex-file version exists (*Legendre_mex.m*). This mex-file copies (nearly) the behavior of *plm.m* *but* it incorporates both the standard and the X-number implementation. Automatically (hence, transparent for the user), the appropriate algorithm

[3]A closed formula for $\overline{P}_{m,m}(t) \propto \sin^m \vartheta$ can be used for accelerating the code.

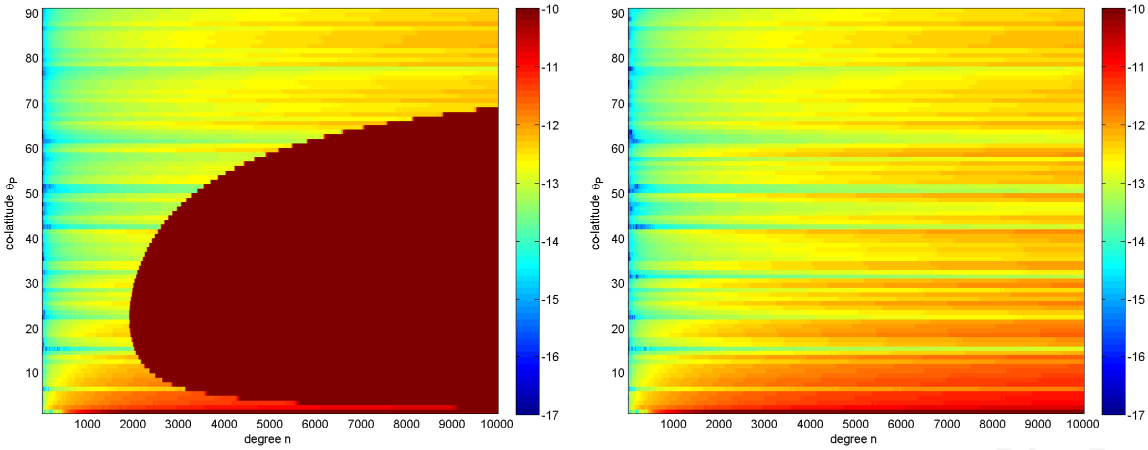


Figure 2.2: Verification of addition theorem (for $\vartheta_P = [0, \frac{\pi}{2}]$, $n = [0, 10\,000]$). Displayed is the base-10-logarithm of the absolute difference between LHS and RHS of relation (2.14). On the left: standard algorithm with breach of addition theorem (dark red area) for high degree/order; on the right: X-numbers extended algorithm without that problem.

is chosen: for $n < 1500$ the faster standard implementation is used, for $n \geq 1500$ (to be on the safe side) the slower – but stable – X-number implementation is selected.

Also, *Legendre_mex.m* supports a mode to calculate all Legendre functions at once (like *legendreP_*.m*). However, be aware of the changed order of Legendre functions due to the optimizations.

Test of Addition Theorem: One of the tests to verify if Legendre functions are computed in a numerically correct way is the so called addition theorem (Xue and Sneeuw, 2006; Sneeuw, 2012)

$$P_n(\cos\psi_{PQ}) = \frac{1}{2n+1}\sum_{m=0}^{n} \overline{P}_{n,m}(\cos\vartheta_P)\overline{P}_{n,m}(\cos\vartheta_Q)\cos m(\lambda_P - \lambda_Q)\,, \tag{2.13}$$

with the spherical distance ψ_{PQ} between points P and Q and their coordinates (λ_i, ϑ_i), $i = P, Q$. If we set $P = Q$, because of $P_n(\cos 0) = 1\ \forall n$, equation (2.13) simplifies to

$$1 = \frac{1}{2n+1}\sum_{m=0}^{n}\left(\overline{P}_{n,m}(\cos\vartheta_P)\right)^2. \tag{2.14}$$

With the standard algorithm, the addition theorem is breached if degree is above $n \approx 1800$ because of an exponent overflow in the numerical calculations. To stabilize the computation, we implemented the X-numbers work-around to catch the overflow and extend the exponent. For verification, we calculated the addition theorem for $\vartheta_P = [0, \frac{\pi}{2}]$ and degree $n = [0, 10\,000]$. The results, displayed in Fig. 2.2, show that the X-number extension works well.

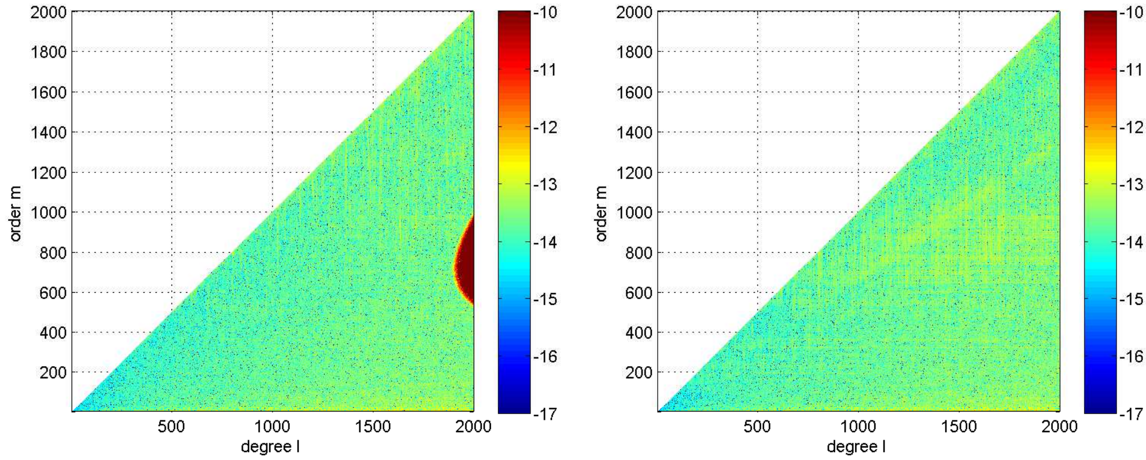


Figure 2.3: Verification of orthogonality. Displayed is the base-10-logarithm of the absolute difference between LHS and RHS of equation (2.16). On the left: standard algorithm with breach of orthogonality (dark red area) for high degree/order; on the right: X-numbers extended algorithm without that problem.

Test of Orthogonality: This is another way to proof the correctness of the normalized Legendre functions which are orthogonal. The orthogonality condition reads for the angle ϑ

$$\frac{1}{2}\int_{0}^{\pi} \overline{P}_{l,m}(\cos\vartheta)\overline{P}_{n,m}(\cos\vartheta)\sin\vartheta\,\mathrm{d}\vartheta = (2-\delta_{m,0})\delta_{l,n}, \tag{2.15}$$

with the Kronecker delta function

$$\delta_{ij} = \begin{cases} 1 & \text{for } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

If we set $l = n$ and rewrite the integral (2.15) as a summation, we get

$$\sum_{i=1}^{N_{\max}} w_i \Big(\overline{P}_{n,m}(\cos\vartheta_i)\Big)^2 = 4 - 2\delta_{m,0}, \tag{2.16}$$

with the maximum degree/order $N_{\max}$, Neumann weights w_i and nodes ϑ_i.

differentiation w. r. t.	$\Lambda_{\eta\xi}$	p_{nm}	α	β
–	1	$\overline{P}_{n,m}(\cos\vartheta)$	$\overline{C}_{n,m}$	$\overline{S}_{n,m}$
r	$-\frac{n+1}{r}$	$\overline{P}_{n,m}(\cos\vartheta)$	$\overline{C}_{n,m}$	$\overline{S}_{n,m}$
ϑ	1	$\frac{\mathrm{d}\overline{P}_{n,m}(\cos\vartheta)}{\mathrm{d}\vartheta}$	$\overline{C}_{n,m}$	$\overline{S}_{n,m}$
λ	1	$m\overline{P}_{n,m}(\cos\vartheta)$	$\overline{S}_{n,m}$	$-\overline{C}_{n,m}$
rr	$\frac{(n+1)(n+2)}{r^2}$	$\overline{P}_{n,m}(\cos\vartheta)$	$\overline{C}_{n,m}$	$\overline{S}_{n,m}$
$r\vartheta$	$-\frac{n+1}{r}$	$\frac{\mathrm{d}\overline{P}_{n,m}(\cos\vartheta)}{\mathrm{d}\vartheta}$	$\overline{C}_{n,m}$	$\overline{S}_{n,m}$
$r\lambda$	$-\frac{n+1}{r}$	$m\overline{P}_{n,m}(\cos\vartheta)$	$\overline{S}_{n,m}$	$-\overline{C}_{n,m}$
$\vartheta\lambda$	1	$m\overline{P}_{n,m}(\cos\vartheta)$	$\overline{S}_{n,m}$	$-\overline{C}_{n,m}$
$\vartheta\vartheta$	1	$\frac{\mathrm{d}^2\overline{P}_{n,m}(\cos\vartheta)}{\mathrm{d}\vartheta^2}$	$\overline{C}_{n,m}$	$\overline{S}_{n,m}$
$\lambda\lambda$	-1	$m^2\overline{P}_{n,m}(\cos\vartheta)$	$\overline{C}_{n,m}$	$\overline{S}_{n,m}$

Table 2.1: Changes of the potential expression V due to partial derivatives w. r. t. spherical coordinates.

2.2 Functionals of the gravity field

The potential V is differentiated w. r. t. the spherical coordinates and the results are combined to achieve different functionals – such as potential values, geoid heights, gravity anomalies, radial derivatives or gravity disturbances – of the gravitational field:

$$\begin{aligned}\mathcal{F}\{V(\lambda,\vartheta,r)\} &= \frac{\mathrm{d}}{\mathrm{d}\eta}\left\{\frac{\mathrm{d}V}{\mathrm{d}\xi}\right\}\\ &= \frac{GM}{R}\sum_{n=0}^{N_{\max}}\sum_{m=0}^{n}\Lambda_{\eta\xi}\left(\frac{R}{r}\right)^{n+1} p_{n,m}\Big(\alpha_{n,m}\cos(m\lambda)+\beta_{n,m}\sin(m\lambda)\Big).\end{aligned} \tag{2.17}$$

In these functionals, partial derivatives w. r. t. $\xi,\eta \in \{\{\ \},r,\vartheta,\lambda\}$ of the basis functions or/and additional factors $\Lambda_{\eta\xi}$ might be involved. Some examples are listed in table 2.1.

2.2.1 Gradient

A very important functional is the gradient of the potential which is required for orbit simulations and gravity field analysis etc. In the 'traditional expressions', the gradient ∇_{t} is determined in a local North oriented frame by the derivatives in spherical coordinates

$$\begin{aligned}\nabla_{\mathrm{t}}V &= \begin{pmatrix}\frac{\partial}{\partial r}\\ -\frac{\partial}{r\partial\vartheta}\\ \frac{\partial}{r\sin\vartheta\partial\lambda}\end{pmatrix}\left\{\frac{GM}{R}\sum_{n=0}^{N_{max}}\left(\frac{R}{r}\right)^{n+1}\sum_{m=0}^{n}\overline{P}_{n,m}(\cos\vartheta)(\overline{C}_{n,m}\cos m\lambda+\overline{S}_{n,m}\sin m\lambda)\right\}\\ &= \frac{GM}{R^2}\sum_{n=0}^{N_{\max}}\left(\frac{R}{r}\right)^{n+1}\sum_{m=0}^{n}\begin{pmatrix}-(n+2)\overline{P}_{n,m}(\cos\vartheta)(\overline{C}_{n,m}\cos m\lambda+\overline{S}_{n,m}\sin m\lambda)\\ -\frac{\mathrm{d}\overline{P}_{n,m}(\cos\vartheta)}{\mathrm{d}\vartheta}(\overline{C}_{n,m}\cos m\lambda+\overline{S}_{n,m}\sin m\lambda)\\ \frac{m\overline{P}_{n,m}(\cos\vartheta)}{\sin\vartheta}(-\overline{C}_{n,m}\sin m\lambda+\overline{S}_{n,m}\cos m\lambda)\end{pmatrix}\end{aligned} \tag{2.18}$$

and the rotation into the Earth-fixed frame (in Cartesian coordinates):

$$\nabla_{\mathrm{e}} V = \underbrace{\begin{pmatrix} \sin\vartheta\cos\lambda & -\cos\vartheta\cos\lambda & -\sin\lambda \\ \sin\vartheta\sin\lambda & -\cos\vartheta\sin\lambda & \cos\lambda \\ \cos\vartheta & \sin\vartheta & 0 \end{pmatrix}}_{R_{\mathrm{e,t}}} \nabla_{\mathrm{t}} V \tag{2.19}$$

Due to the term $\frac{1}{\sin\vartheta}$, the expression (2.18) will become numerically singular, if the calculation point is (very) close to the poles.

To overcome this singularity, the "method of linear combination"

$$\nabla V = \frac{GM}{2R^2} \sum_{n=0}^{N_{\max}} \left(\frac{R}{r}\right)^{n+2} \sqrt{\frac{2n+1}{2n+3}} \sum_{m=0}^{n} \overline{C}_{n,m} \begin{pmatrix} c^-_{nm} - c^+_{nm} \\ -s^-_{nm} - s^+_{nm} \\ -2c^0_{nm} \end{pmatrix} + \overline{S}_{n,m} \begin{pmatrix} s^-_{nm} - s^+_{nm} \\ c^-_{nm} + c^+_{nm} \\ -2s^0_{nm} \end{pmatrix} \tag{2.20}$$

with

$$\begin{Bmatrix} c^-_{nm} \\ s^-_{nm} \end{Bmatrix} = \sqrt{(n-m+1)(n-m+2)(1+\delta_{1m})}\,\overline{P}_{n+1,m-1}(\cos\vartheta) \begin{Bmatrix} \cos((m-1)\lambda) \\ \sin((m-1)\lambda) \end{Bmatrix} \tag{2.21}$$

$$\begin{Bmatrix} c^+_{nm} \\ s^+_{nm} \end{Bmatrix} = \sqrt{(n+m+1)(n+m+2)(1+\delta_{0m})}\,\overline{P}_{n+1,m+1}(\cos\vartheta) \begin{Bmatrix} \cos((m+1)\lambda) \\ \sin((m+1)\lambda) \end{Bmatrix} \tag{2.22}$$

$$\begin{Bmatrix} c^0_{nm} \\ s^0_{nm} \end{Bmatrix} = \sqrt{(n-m+1)(n+m+2)}\,\overline{P}_{n+1,m}(\cos\vartheta) \begin{Bmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{Bmatrix}. \tag{2.23}$$

can be used (see Mayer-Gürr (2006), respectively Ilk (1983)), which is implemented in the routine *gradpshs.m*.

Alternatively, the recursion could be modified according to Haines (1988) to determine an auxiliary function

$$a_{n,m}(\cos\vartheta) := \frac{m\overline{P}_{n,m}(\cos\vartheta)}{\sin\vartheta}, \tag{2.24}$$

which would remove the singularity of the gradient. This idea is implemented in some of the single point synthesis functions, for example in the routine *sph_gradshs.m*.

2.2.2 Tensor

In an analogous manner to the gradient, the tensor is expressed in the traditional representation in spherical coordinates and the local north oriented frame

$$
\begin{aligned}
V_{ZZ} &= V_{rr} && \text{(2.25a)}\\
V_{XX} &= \frac{1}{r}V_r + \frac{1}{r^2}V_{\vartheta\vartheta} && \text{(2.25b)}\\
V_{YY} &= \frac{1}{r}V_r + \frac{1}{r^2\tan\vartheta}V_\vartheta + \frac{1}{r^2\sin^2\vartheta}V_{\lambda,\lambda} && \text{(2.25c)}\\
V_{XY} &= \frac{1}{r^2\sin\vartheta}V_{\vartheta\lambda} - \frac{\cos\vartheta}{r^2\sin^2\vartheta}V_\lambda && \text{(2.25d)}\\
V_{XZ} &= \frac{1}{r^2}V_\vartheta - \frac{1}{r}V_{r\vartheta} && \text{(2.25e)}\\
V_{YZ} &= \frac{1}{r^2\sin\vartheta}V_\lambda - \frac{1}{r\sin\vartheta}V_{r\lambda} && \text{(2.25f)}
\end{aligned}
$$

(Koop, 1993), and rotated towards the Earth-fixed frame

$$\nabla_{\mathrm{e}}^2 V = R_{\mathrm{e,t}} \nabla_{\mathrm{t}}^2 V R_{\mathrm{e,t}}^\top . \tag{2.26}$$

Parts of the singularity could be solved by the auxiliary function again, but for the remaining components (V_{YY}, V_{XY}) the method of linear combination would be necessary, which is discussed in Petrovskaya and Vershkov (2006).

2.3 Location of calculating points

The routines of the spherical harmonic synthesis differ in the location of the calculation points:

- **grid**:
  The functional of the gravitational field is calculated on a regular grid on the sphere (or in the orbit). The spacing in longitude and latitude direction might differ, but both vectors are usually equidistant. The vectors might have different lengths.
- **pointwise**
  Arbitrary locations on the sphere or in the orbit (e. g. along the satellite track) are used in the synthesis. The vectors of longitude, latitude and radius must have the same dimension, which is checked by *checkcoor.m* (in the package UBERALL).
- **single point**
  In some applications such as orbit integration only one time points is used in the synthesis routines. In orbit integration multiple calls are necessary, so that the synthesis is often optimized for this case.

2.4 Sorting and storing

Spherical harmonic functions or coefficients, Legendre functions and their derivatives can be arranged in different ways. For the functions, the sorting can make a difference in computation time, but for the coefficients it is only a matter of convention or convenience. There are multiple routines in the SHBUNDLE for reordering from on format to another format:

clm-format: One of the standard ways to store spherical harmonic coefficients is the indexed column-vector-format (abbreviated: clm-format)[4]:

$$\left(n, m, \overline{C}_{n,m}, \overline{S}_{n,m}, \left[\sigma_{\overline{C}_{n,m}}, \sigma_{\overline{S}_{n,m}}\right]\right) \tag{2.27}$$

In this format, the first column represents the degree n, the second column contains the order m (with n, m integer numbers), followed by the coefficients $\overline{C}_{n,m}, \overline{S}_{n,m}$ and – if given – their standard deviations $\sigma_{\overline{C}_{n,m}}, \sigma_{\overline{S}_{n,m}}$.

Colombo ordering is a special case of the clm-format. Here, the coefficients are sorted first by order m, then by degree n.

Klm-Format is a variation of the clm-format for the compact notation (2.8) in 3 or 4 columns. The sine-coefficients are notated down by negative orders, and the coefficients are sorted firstly with respect to degree and then the order:

$$\begin{pmatrix} 0 & 0 & \overline{C}_{0,0} & \sigma_{\overline{C}_{0,0}} \\ 1 & -1 & \overline{S}_{1,1} & \sigma_{\overline{S}_{1,1}} \\ 1 & 0 & \overline{C}_{1,0} & \sigma_{\overline{C}_{1,0}} \\ 1 & 1 & \overline{C}_{1,1} & \sigma_{\overline{C}_{1,1}} \\ 2 & -2 & \overline{S}_{2,2} & \sigma_{\overline{S}_{2,2}} \\ 2 & -1 & \overline{S}_{2,1} & \sigma_{\overline{S}_{2,1}} \\ 2 & 0 & \overline{C}_{2,0} & \sigma_{\overline{C}_{2,0}} \\ 2 & 1 & \overline{C}_{2,1} & \sigma_{\overline{C}_{2,1}} \\ 2 & 2 & \overline{C}_{2,2} & \sigma_{\overline{C}_{2,2}} \\ \vdots & & & \vdots \\ N_{\max} & N_{\max} & \overline{C}_{N_{\max},N_{\max}} & \sigma_{\overline{C}_{N_{\max},N_{\max}}} \end{pmatrix} \tag{2.28}$$

The 4th column can be neglected, if the information is not available.

[4]The name referrers to the degree (l) instead of n, the order (m) and the word coefficients.

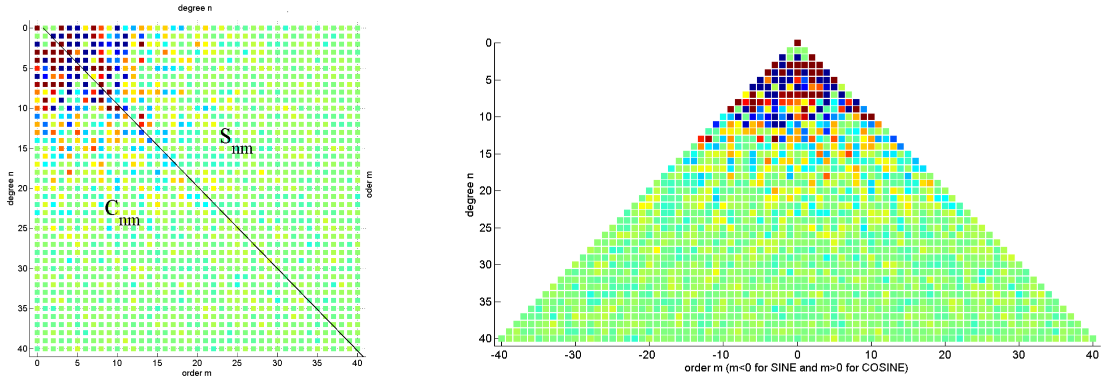


Figure 2.4: Examples of spherical harmonic coefficients in the `|C\S|`-format (left hand side) and the `/S|C\`-format (right hand side).

`|C\S|`-format: A well-known arrangement is the quadratic `|C\S|`-format of spherical harmonic coefficients:

$$
\begin{pmatrix}
\overline{C}_{0,0} & \overline{S}_{1,1} & \overline{S}_{2,1} & \overline{S}_{3,1} & \dots & & \overline{S}_{N_{\max},1} \\
\overline{C}_{1,0} & \overline{C}_{1,1} & \overline{S}_{2,2} & \overline{S}_{3,2} & \dots & & \overline{S}_{N_{\max},2} \\
\overline{C}_{2,0} & \overline{C}_{2,1} & \overline{C}_{2,2} & \overline{S}_{3,3} & \dots & & \overline{S}_{N_{\max},3} \\
\overline{C}_{3,0} & \overline{C}_{3,1} & \overline{C}_{3,2} & \overline{C}_{3,3} & \dots & & \overline{S}_{N_{\max},4} \\
\overline{C}_{4,0} & \overline{C}_{4,1} & \overline{C}_{4,2} & \overline{C}_{4,3} & \overline{C}_{4,4} & \dots & \overline{S}_{N_{\max},5} \\
\vdots & & & & & \ddots & \\
\overline{C}_{N_{\max},0} & \overline{C}_{N_{\max},1} & \overline{C}_{N_{\max},2} & \overline{C}_{N_{\max},3} & \overline{C}_{N_{\max},4} & \dots & \overline{C}_{N_{\max},N_{\max}}
\end{pmatrix} . \tag{2.29}
$$

In this format, the cosine terms are ordered in the lower triangle, where the column refers to the order m and row is related to the degree n. The sine terms are rotated by $90°$, and the order $m = 0$ is removed (see Fig. 2.4).

`/S|C\`-format: In the popular `/S|C\`-format the sine-coefficients are flipped from left to right, to obtain a triangular matrix which is completed by zeros (see Fig. 2.4).

$$
\begin{pmatrix}
 & & & & & \overline{C}_{0,0} & & & & \\
 & \mathbf{0} & & & \overline{S}_{1,1} & \overline{C}_{1,0} & \overline{C}_{1,1} & & \mathbf{0} & \\
 & & & \overline{S}_{2,2} & \overline{S}_{21} & \overline{C}_{2,0} & \overline{C}_{2,1} & \overline{C}_{2,2} & & \\
 & & \overline{S}_{3,3} & \overline{S}_{32} & \overline{S}_{3,1} & \overline{C}_{3,0} & \overline{C}_{3,1} & \overline{C}_{3,2} & \overline{C}_{3,3} & \\
 & \iddots & & & & \vdots & & & \ddots & \\
\overline{S}_{N_{\max},N_{\max}} & & \dots & & \overline{S}_{N_{\max},1} & \overline{C}_{N_{\max},0} & \overline{C}_{N_{\max},1} & \dots & & \overline{C}_{N_{\max},N_{\max}}
\end{pmatrix} . \tag{2.30}
$$

The functions *cs2sc.m* and *sc2cs.m* switch the arrangement between the two matrix formats.

vector-formats: In several routines, a vectorial arrangement is preferred for coefficients or Legendre functions. The **degree-order** format is given by

$$
\begin{aligned}
&\left(\overline{C}_{0,0} \;\; \overline{C}_{1,0} \;\; \overline{C}_{1,1} \;\; \overline{C}_{2,0} \;\; \overline{C}_{2,1} \;\; \overline{C}_{2,2} \;\; \overline{C}_{3,0} \;\; \overline{C}_{3,1} \;\; \dots \;\; \overline{C}_{N_{\max},N_{\max}}\right)^{\top} \\
&\left(\overline{S}_{0,0} \;\; \overline{S}_{1,0} \;\; \overline{S}_{1,1} \;\; \overline{S}_{2,0} \;\; \overline{S}_{2,1} \;\; \overline{S}_{2,2} \;\; \overline{S}_{3,0} \;\; \overline{S}_{3,1} \;\; \dots \;\; \overline{S}_{N_{\max},N_{\max}}\right)^{\top}
\end{aligned} \tag{2.31}
$$

while the **order-degree** sorting looks like

$$\begin{aligned}&\begin{pmatrix}\overline{C}_{0,0} & \overline{C}_{1,0} & \overline{C}_{2,0} & \overline{C}_{3,0} & \dots & \overline{C}_{N_{\max},0} & \overline{C}_{1,1} & \overline{C}_{2,1} & \dots & \overline{C}_{N_{\max},N_{\max}}\end{pmatrix}^{\top} \\ &\begin{pmatrix}\overline{S}_{0,0} & \overline{S}_{1,0} & \overline{S}_{2,0} & \overline{S}_{3,0} & \dots & \overline{S}_{N_{\max},0} & \overline{S}_{1,1} & \overline{S}_{2,1} & \dots & \overline{S}_{N_{\max},N_{\max}}\end{pmatrix}^{\top}.\end{aligned} \tag{2.32}$$

For consistency, the order $m = 0$ in the sine-coefficients is kept, although they are all zero by definition!

The (solid) spherical harmonics and Legendre functions – e. g. in the routines *legendreP.m* – are arranged in the degree-order sorting for the rows, while the columns contain the different positions in space. The functions *legendreP_mex.m* is optimized w. r. t. computation time, where the rows of the output are sorted in the **order-degree** format.

The indices for a particular degree and order can be found in the additional output of a degree-order matrix in the same routine *legendreP.m*, but also via the functions *legendreIndex.m* and *sortLegendre.m*.

3 Some specific functions

3.1 Complex representations of SHS

The spherical harmonic synthesis (2.7) is a two-dimensional and generalized Fourier series on the sphere. Like in the case of one-dimensional signal analysis, the same series can be expressed also in a complex notation:

$$V(\lambda,\vartheta,r)=\frac{GM}{R}\sum_{n=0}^{N_{\max}}\sum_{m=-n}^{n}\left(\frac{R}{r}\right)^{n+1}\hat{Z}_{n,m}\hat{Y}_{n,m}(\lambda,\vartheta). \tag{3.1}$$

Please consider, that the complex representation differs often by the normalization of spherical harmonics, which are here noted down now by $\hat{Y}_{n,m}(\lambda,\vartheta)$ instead of $\bar{Y}_{n,m}(\lambda,\vartheta)$, and the coefficients $\hat{Z}_{n,m}$ must be consistent with the chosen normalization.

There are several "complex normalizations" of spherical harmonics, and the SHBUNDLE offers at the moment only order-depending scaling factors in the function *LeNorm.m*. The so-called "geodetic complex normalization", which is used in the routines *cpx2realmat.m* and *real2cpxmat.m* and also in the upcoming filter-bundle, is given by

$$\hat{Y}_{n,\pm m}(\lambda,\vartheta)=\overline{P}_{n,|m|}(\cos\vartheta)\mathrm{e}^{\pm\imath m\lambda}\cdot\begin{cases}(-1)^m & m>0\\ \frac{1}{\sqrt{2}} & m=0\\ 1 & m<0\end{cases} \tag{3.2}$$

and the version implemented in MATHEMATICA is given by

$$\hat{Y}^{M}_{n,\pm m}(\lambda,\vartheta)=\overline{P}_{n,|m|}(\cos\vartheta)\mathrm{e}^{\pm\imath m\lambda'}\cdot\begin{cases}\frac{(-1)^m}{2\sqrt{2\pi}} & m>0\\ \frac{1}{2\sqrt{\pi}} & m=0\\ \frac{1}{2\sqrt{2\pi}} & m<0.\end{cases} \tag{3.3}$$

Please note, that alternative normalizations are usually implemented by a modified recursion scheme instead of a final multiplication of functions and factors, and that a degree-dependent factor is currently not considered.

The sum of two exponential functions with the same complex frequency – or a sine- and cosine-term of the same frequency – can always be summarized to one function by determining its amplitude and phase. In (Devaraju, 2015) an amplitude-phase representation of the spherical harmonic synthesis is introduced

$$V(\lambda,\vartheta,r)=\frac{GM}{R}\sum_{n=0}^{N_{\max}}\sum_{m=-n}^{n}\left(\frac{R}{r}\right)^{n+1}\hat{A}_{n,m}\hat{Y}_{n,m}(\lambda,\vartheta)\mathrm{e}^{\imath m\phi_{n,m}}, \tag{3.4}$$

and the values $\{\hat{A}_{n,m},\phi_{n,m}\}$ are calculated by the routine *shspkamph.m* from the original coefficients. The amplitude-phase representation od SHS is rarely used in geodesy, but it is helpful in understanding of (an-isotropic or inhomogeneous) filtering on the sphere.

3.2 Wigner-d-functions and rotations

To match the standard gravity field products of the geoscience calculation centers, all Legendre functions in the SHBUNDLE are implemented in the geodetic normalization, which is described in section 2.1. Alternative normalizations must be derived by multiplication with order-dependent factors after the recursive calculation.

The alternatives normalizations are helpful for the rotation of spherical harmonics, either for expressing the gravity field quantities in another coordinate system, or for avoiding the singularity of the gradient in polar regions (Sneeuw, 1991; Kostelec and Rockmore, 2003; Devaraju, 2015; Fukushima, 2017). The concept is based on the SO(3) group and the relationship

$$\hat{Y}_{n,m}(\lambda', \vartheta') = \mathrm{e}^{-\imath m\gamma} \sum_{k=-n}^{n} d_{n,m,k}(\beta)\mathrm{e}^{-\imath k\alpha}\hat{Y}_{n,k}(\lambda, \vartheta) \tag{3.5}$$

- (α, β, γ): rotation angles of an Euler-rotation,
- (λ, ϑ): coordinates in the original system,
- (λ', ϑ'): coordinates in the rotated system,
- $d_{n,m,k}(..)$: Wigner-d-function[1] of degree n and the integer orders (m, k),
- $\hat{Y}_{n,m}(..)$: spherical harmonics in a complex form and normalization.

In geodesy, this relationship is often used in an implicit way, as the well-known inclination functions are – apart from normalization – a product of the Wigner-d-function $d_{n,m,k}(..)$ and the Legendre functions $\hat{Y}_{n,k}(..)$ (Sneeuw, 1991).

The rotation of spherical harmonics by Wigner-d-functions and the inclination functions justifies another toolbox. There are several algorithms via series expansion, Fourier method, convolution, or recursive algorithms (Sneeuw, 1991; Gooding and Wagner, 2010). Some methods generate only the function of single set of orders (m, k), others are restricted to a scalar angle β. In (Fukushima, 2017) a modified algorithm is presented for the "rectangular rotation" for a fixed value $\beta = \frac{\pi}{2}$, which is then re-formulated for a X-number version of Wigner-d-functions for high degree expansions.

Nevertheless, two routines are included in the SHBUNDLE. The routine *wigner_all.m* provides all Wigner-d-functions – or the inclination functions – up to maximum degree $N_{\max}$, but only for a scalar angle β. The algorithm is based on (Kostelec and Rockmore, 2003), but modified for all integer degrees and orders in one call. The output enables the rotation of one or more spherical harmonic functions in the complex representation according to formula (3.5), and the result can be converted to real spherical harmonic afterwards.

The spherical harmonic synthesis is linear w.r.t. the coefficients and basis functions. Hence, the rotation can be applied on the spherical harmonic functions or on the coefficients. Latter concept is implemented for real- or complex-valued coefficients in the function *rotate_shc.m*.

In Fig. 3.1 the Wigner-d-functions are acting in the left panel on the spherical harmonic functions and in the right panel on the spherical harmonic coefficients. Both result are verified – up to numerical effects – by the rotation of the coordinate system in the middle figure.

[1]Unfortunately, there is no standard notation for the Wigner-d-functions. Some authors use other letters for degree n or orders (m, k), or note down the degree as an upper index.

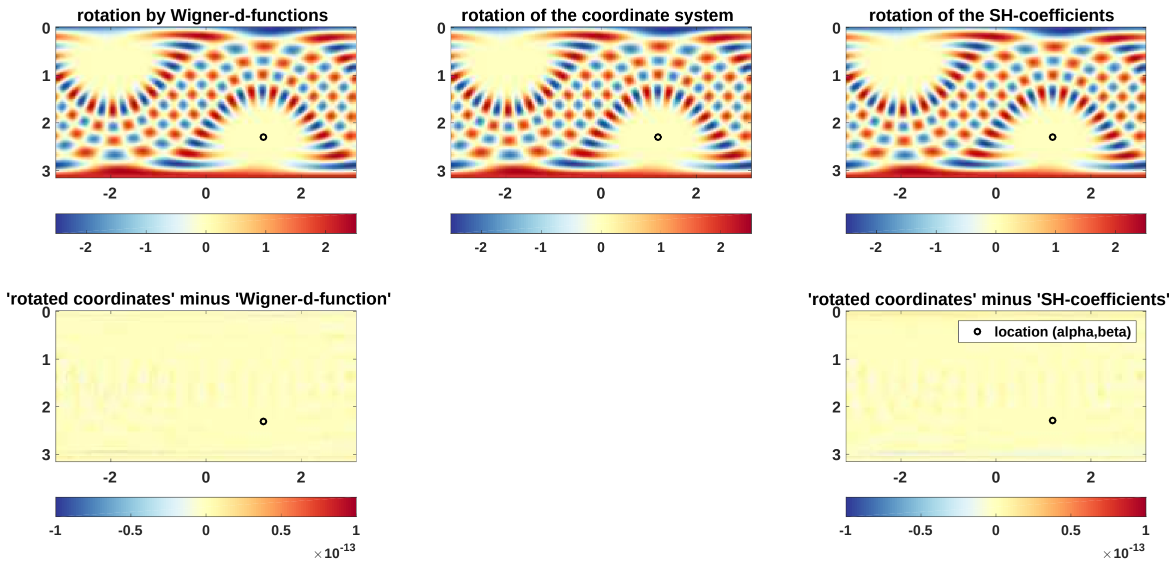


Figure 3.1: Rotation of the spherical harmonic function $\overline{P}_{15,11}(\cos\vartheta)\cos(11\lambda)$
In the left figure, the rotation is performed by the Wigner-d-functions acting on the function and in the right figure by acting on the coefficients and a subsequent standard SHS. The figure in the middle is the reference by rotating the coordinate system and evaluating afterwards.

A rotation of the coefficients instead of coordinates or functions has some benefits:

- By rotating the representation of the gravity field – but not the field values –, the singularity of the gradient at the pole is irrelevant. In (Fukushima, 2017) it is suggested, to prepare two sets of spherical harmonic coefficients for the evaluation of the gravity field, and to select the adequate one depending on the location of the evaluation point.
- For regional analysis, alternative basis function of the gravity field representation are developed and investigated since many years. These alternatives are usually localizing basis function, which model only data in a small region of interest and their function value tends to zero outside. All alternative basis functions have – due to the completeness of the spherical harmonic system – a spectral representation, i.e. a corresponding set of spherical harmonic coefficients. The rotation of the coefficients generates a copy of the localizing basis functions and their properties in other regions. This is in particular helpful, when the construction of the localizing basis function requires high computational effort like the eigenvalue problem for the Slepian basis functions.

3.3 Spherical Slepian basis functions

Spherical Slepian functions are localizing basis functions $g_\alpha(\lambda, \vartheta)$, which are constructed for a given sub-region $\mathcal{R}$ of the sphere Σ in the spectral domain, i.e. by a spherical harmonic synthesis. Those basis functions which are "optimally concentrated" within the region

$$\lambda_\alpha = \frac{\iint_{\mathcal{R}} {g_\alpha}^2(\lambda, \vartheta) \sin\vartheta \mathrm{d}\lambda \mathrm{d}\vartheta}{\iint_{\Sigma} {g_\alpha}^2(\lambda, \vartheta) \sin\vartheta \mathrm{d}\lambda \mathrm{d}\vartheta} = \max \tag{3.6}$$

are primarily called Slepian[2] functions. In addition, the resulting basis functions are mutually orthogonal on the complete sphere Σ and within the sub-region $\mathcal{R}$.

F. J. Simons, F.A. Dahlen and M. Wieczorek made the Slepian approach popular in geoscience by a series of articles (Wieczorek and Simons (2005); Simons and Dahlen (2006); Simons et al. (1961), etc.) and they also published their toolboxes:

- SLEPIAN_ALPHA written in MATLAB by F. J. Simons `geoweb.princeton.edu/people/simons/software.html` (also available on `https://github.com`)
- PYSHTOOLS 4.9 written in PYTHON by M. Wieczorek and M. Meschede `https://shtools.oca.eu/shtools/public/index.html`

The SHBUNDLE is currently restricted to a simple but important case, when the sub-region $\mathcal{R}$ is a spherical cap (around an arbitrary center). Slepian functions within (small) spherical caps can be used to analyze direction-dependent signals on the sphere.

3.3.1 Theory of Slepian basis functions

In (Simons and Dahlen, 2006), the spherical harmonic functions are introduced via a real representation, but with an alternative normalization. In addition, the normalization factors are distributed between Legendre functions and spherical harmonic functions, where only the latter version is shown here. Using the routines of the SHBUNDLE an order-dependent factor

$$\check{Y}_{n,m} = \frac{(-1)^m}{\sqrt{4\pi}} \overline{P}_{n,m}(\cos\vartheta) \cdot \begin{cases} \cos m\lambda & m \geq 0 \\ \sin m\lambda & m < 0 \end{cases} \tag{3.7}$$

is necessary for conversion of spherical harmonics in both toolboxes. Experiments with the routine `ptoslep.m` in SLEPIAN_ALPHA indicate, that for the Slepian functions within a spherical cap the "standard" normalization (2.6) is used in the synthesis. The theory is explained in the following with the "$\check{Y}_{n,m}$-notation" to highlight the possibility of an alternative normalization in the Slepian approach in other toolboxes[3].

[2]The name honors the mathematician David Slepian (1923–2007), who investigated signals with spatial and spectral concentration under the label *prolate spheroidal wave functions* in several articles (e.g. Slepian and Pollak (1961)).

[3]The different normalization might play a role when the coefficients are calculated via the Grünbaum matrix (section 3.3.2), but in the general method the result should be self-consistent to our understanding.

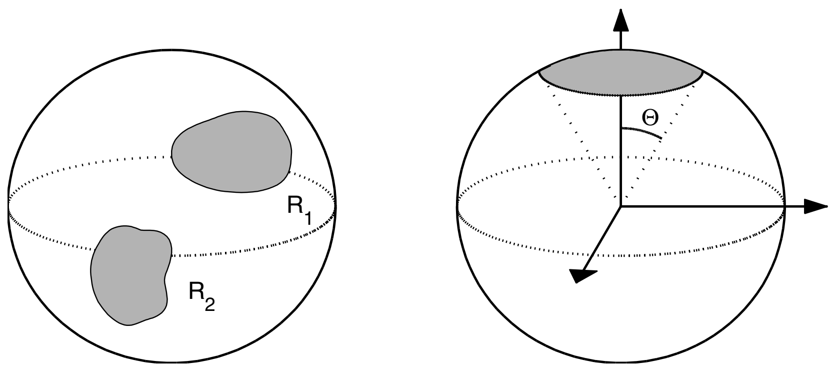


Figure 3.2: Examples of sub-regions $\mathcal{R}$ on the sphere: The left panel shows two distinct regions of arbitrary shape and location, the right panel shows a spherical cap around the North pole (Simons et al., 1961).

A gravitational potential is modeled in the Slepian approach by the linear combination

$$V(\lambda,\vartheta,r) = \frac{GM}{R}\sum_{\alpha=1}^{A} s_\alpha \sum_{n=0}^{N_{\max}} \left(\frac{R}{r}\right)^{n+1} \sum_{m=-n}^{n} \check{h}^{\alpha}_{n,m}\check{Y}_{n,m}(\lambda,\vartheta). \tag{3.8}$$

with the coefficients/scaling factors s_α. On the reference surface ($r = R$), each Slepian basis function is generated by a spherical harmonic synthesis

$$g_\alpha(\lambda,\vartheta) = \sum_{n=0}^{N_{\max}} \sum_{m=-n}^{n} \check{h}^{\alpha}_{n,m}\check{Y}_{n,m}(\lambda,\vartheta). \tag{3.9}$$

A series expansion up to degree $n = N_{\max}$ enables $(N_{\max}+1)^2$ independent linear combinations, which is equivalent to the amount of spherical harmonic coefficients in the synthesis. The Slepian functions can be grouped – by their corresponding eigenvalues – into basis functions with high or low concentration within the sub-region $\mathcal{R}$. Using only highly concentrated Slepian functions might significantly reduce the amount of basis functions in the modelling.

The determination of the coefficients requires for an area of arbitrary shape (e.g. water basin, country, continent etc.) the following steps (Simons and Dahlen, 2006; Simons et al., 1961):

- The quadrature of all possible products of spherical harmonic functions

$$D_{n,m,l,k} = \iint_{\mathcal{R}} \check{Y}_{n,m}(\lambda,\vartheta)\check{Y}_{l,k}(\lambda,\vartheta)\sin\vartheta \mathrm{d}\vartheta\mathrm{d}\lambda$$

  up to maximum degree $N_{\max}$.

- The arrangement of these integrals in one symmetric matrix:

$$\underline{\mathbf{D}} = \begin{pmatrix} D_{0,0,0,0} & \dots & D_{0,0,N_{\max},N_{\max}} \\ D_{N_{\max},N_{\max},0,0} & \dots & D_{N_{\max},N_{\max},N_{\max},N_{\max}} \end{pmatrix}.$$

- The solution of the eigenvalue problem

$$\underline{\mathbf{D}}\check{\mathbf{h}}^{\alpha} = \Lambda_{\alpha}\check{\mathbf{h}}^{\alpha}$$

  where the eigenvalues Λ_α are – due to theory – restricted to the interval $[0, 1]$. Eigenvalue of magnitude $\Lambda_\alpha \approx 1$ indicate a high concentration of the basis function within the sub-region $\mathcal{R}$.

- The selection and re-ordering of the coefficients from the eigenvectors $\check{\mathbf{h}}^{\alpha}$, and their spherical harmonic synthesis for the determination of the Slepian basis functions.

3.3.2 Slepian basis functions within a spherical cap

When the region is a spherical cap with radius Θ, the coefficients can be found by a much simpler eigenvalue problem $\underline{\mathbf{T}}\check{\mathbf{h}}^{\alpha} = \chi_\alpha \check{\mathbf{h}}^{\alpha}$ of the Grünbaum matrix $\underline{\mathbf{T}}$. The Grünbaum matrix is tri-diagonal and symmetric, and the entries are found by closed formulas (cf. Simons and Dahlen (2006)):

$$T_{i,i+1} = T_{i+1,i} = -n(n+1)\cos\Theta$$

$$T_{i,i} = [n(n+2) - n(N_{\max}+2)]\sqrt{\frac{(n+1)^2 - m^2}{(2n+1)(2n+3)}}$$

The eigenvectors are determined in the routine *evp_gruenbaum.m* for a spherical cap around the North pole. The coefficients are calculated per order m and stored in a sparse matrix for memory and readability. By default, the result is sorted according to the eigenvalues, so that the functions with the smaller indices are best concentrated within the sub-region $\mathcal{R}$. Please note, that only the eigenvectors $\check{\mathbf{h}}^{\alpha}$ are equal to the previous method[4], but not the eigenvalues $\chi_\alpha \neq \Lambda_\alpha$. In fact, the highly concentrated Slepian functions have now the smallest, even negative Grünbaum eigenvalues χ_α. The concentration decreases for increasing Grünbaum eigenvalues χ_α, but the “last good basis function” is difficult to determine from the Grünbaum eigenvalues alone.

At the moment, the Slepian basis functions must be selected manually by entering their indices and the output of the previous routine into the function *getSlepianSHC.m*. Here, the eigenvectors are rearranged to a `|C\S|`-format per Slepian basis function. It is also possible, to rotate the spherical cap to another location by inserting the Euler angles (and the internal call of the routine *rotate_shc.m*). Figure 3.3 illustrates a set of Slepian functions after rotation, where the functions in the upper panels are highly concentrated within the spherical cap and the lower panels present basis functions with low concentration.

Specific disclaimer – Slepian basis functions

The Slepian approach should be handled with care and it might require a revision. In a comparison of SHBUNDLE and SLEPIAN_ALPHA of F. J. Simons, the following aspects should be kept in mind:

[4]The identical eigenvectors $\check{\mathbf{h}}^{\alpha}$ are a consequence of the fact, that the 2 matrices commute, i.e. $\underline{\mathbf{T}}\,\underline{\mathbf{D}} = \underline{\mathbf{D}}\,\underline{\mathbf{T}}$.

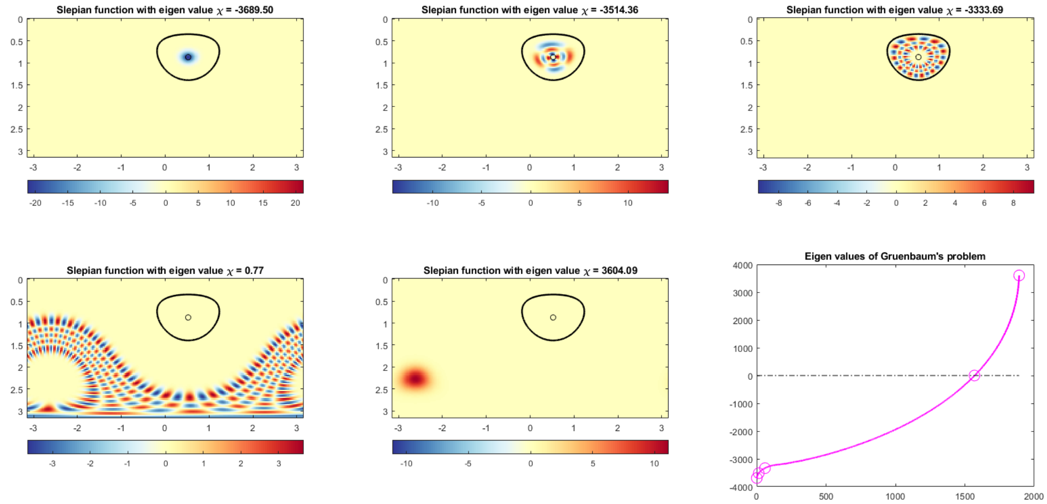


Figure 3.3: Examples of Slepian functions within a spherical cap of radius $\Theta = \frac{\pi}{6}$ and expanded up to $N_{\max} = 60$. The basis functions are rotated by the angles $\alpha = \frac{\pi}{6}$, $\beta = \frac{5\pi}{18}$ and $\gamma = \frac{11\pi}{36}$ to a new location. The bold "circle" indicates the boundary of the spherical cap, the inner circle is the location of the center.

- The Grünbaum matrix has the same entries in both toolboxes, but some corresponding eigenvectors differ by a factor (-1). In the routine `grunbaum.m` of the toolbox SLEPIAN_ALPHA, a first spherical harmonic synthesis is applied after the eigenvalue problem. If the Slepian function along the meridian $\lambda = 0$ starts with a negative lobe, the eigenvector is replaced by its negative counterpart. This behaviour is documented in the code, but not explained in (Simons and Dahlen, 2006). This manipulation is not implemented in the SHBUNDLE yet, and so the coefficients s_α in formula (3.8) might differ by a minus sign from Simons's results.

- An alternative normalization of spherical harmonics is introduced (Simons et al., 1961) and implemented in SLEPIAN_ALPHA. At least for the Slepian functions within a spherical cap this is not considered, as the routine `plm2xyz.m` uses the built-in Legendre function of MATLAB. In combination with additional factors, the resulting Legendre functions are equivalent to the form (2.6).

- To obtain identical results in our tests, it is necessary to replace the rotation angle γ – for changing the azimuth direction in the rotated coordinate system – by its negative counterpart in one of the toolboxes.

Please feel free to contact us for remarks, suggestions or improvements of this initial Slepian approach via bundle@gis.uni-stuttgart.de.

4 List of functions

The following list presents all routines in an alphabetically order. For a better overview, a color scheme is added for Legendre functions, spherical harmonic synthesis & analysis and reordering & indices routines.

checkshformat.m	checks the format of (possible) spherical harmonic coefficients and returns a string for the type and the maximum degree.
clm2klm.m	converts spherical coefficients ordered in clm-format to klm-format (inverse of *klm2clm.m*).
clm2sc.m	converts a list of coefficients in clm-format to `/S|C\`-format and – if given – does the same with the standard deviations.
covord.m	converts a covariance matrix of a spherical harmonic degree expansion into a structure of four variables with each variable containing cells.
cpx2realcov.m	generates a matrix the size of a spectral covariance matrix for a given spherical harmonic degree.
cpx2realmat.m	generates a matrix operator that converts complex spherical harmonics to real spherical harmonics.
cpx2realsh.m	converts complex spherical harmonic coefficients to real spherical harmonic coefficients.
cs2sc.m	rearranges spherical harmonic coefficients from the `|C\S|`-format (2.29) to the `/S|C\`-format (2.30) (inverse of *sc2cs.m*).
cs2vec.m	rearranges spherical harmonic coefficients from the `|C\S|`-format (2.29) to the vector-format (2.31) (inverse of *vec2cs.m*).
cssc2clm.m	converts `|C\S|`- and `/S|C\`-format matrices and degree variance information to clm-format with Colombo ordering (2.27).
degcorr.m	determines the degree correlation between two or more models of spherical harmonic coefficients.
degreeinfo.m	determines statistical information (like: degree-RMS, variances, cumulative RMS,...) for a set of spherical harmonic coefficients.
diffLegpol.m	calculates the unnormalized Legendre polynomials $P_n(t)$ with $t = \cos\vartheta$ including the 1st, 2nd and 3rd derivatives w. r. t. the argument t.

eigengrav.m	provides the isotropic spectral transfer (or: eigenvalues) to switch between the gravity related quantities: • gravitational potential [m^2/s^2] • geoid height [m] • gravity disturbances [mGal] • gravity anomaly [mGal] • 2^{nd}radial derivative of the field [E] • slope [arcsec] • equivalent water thickness [m] • surface mass density • coefficient dependent.
evp_gruenbaum.m	solves the eigenvalue problem of Grünbaum for the Slepian basis functions which are concentrated within a spherical cap.
getSlepianSHC.m	rearranges the coefficients of the Slepian basis functions – generated by *evp_gruenbaum.m* – into standard `|C\S|`-format per basis function.
globalmean.m	computes the mean of a field on a unit sphere which is given in spectral or spatial domain.
globalpower.m	computes the power of a field on a unit sphere which is given in spectral or spatial domain.
gradpshs.m	determines the gradient of a field given in spherical harmonic coefficients for a vector of locations (i. e. pointwise) via the linear combination (2.20).
gsha.m	calculates the spherical harmonic coefficients for a gridded global field by • least squares • weighted least squares • approximate quadrature • first Neumann method • second Neumann method • block mean values.
gshs2ptfun.m	performs the spherical harmonic synthesis of the spectrum of a two-point function, for example, covariance functions, smoothing operators and such.
gshs_.m	calculates a spherical harmonic synthesis for a global and regular grid. Output selection: see *eigengrav.m*.
gshscov.m	propagates the covariance information of a spherical harmonic field to the spatial domain based on the GSHS approach.
gshs_grid.m	calculates a global spherical harmonic synthesis for any grid, which is defined by longitude and latitude vectors (with different dimensions). Output selection: see *eigengrav.m*.

gshs_ptw.m	calculates a global spherical harmonic synthesis pointwise (e. g. along an orbit), where the longitude and latitude vectors must have the same dimensions. Output selection: see *eigengrav.m*.
iplm.m	integrates the Legendre function via a modified recursion formula for $\int \overline{P}_{n,m}(\cos\vartheta)\sin\vartheta \mathrm{d}\vartheta$.
iplmquad.m	provides the kernel for numerical integration of Legendre functions by *intpnm.m*.
kaula.m	estimates the signal content of spherical harmonic coefficients by Kaula's rule of thumb.
klm2clm.m	converts spherical coefficients ordered in klm-format to clm-format (inverse of *clm2klm.m*)
Legendre0.m	calculates the associated Legendre functions (in geodetic normalization) at the equator with exact zeros (usable for rotations of spherical harmoncis).
legendreIndex.m	determines the index for selecting Legendre functions in degree-order sorting. Degree or order can be a vector, while the second value must be scalar.
Legendre_mex.m	computes the covariance function in the spatial domain for a given point in space. For degree/order $N_{\max} < 1500$ the fast standard recursion is used, for $N_{\max} \geq 1500$, the X-number enhanced algorithm is used for numerical stability (tested until degree/order $N_{\max} = 30\,000$).
legendreP.m	calculates the associated Legendre functions (in geodetic normalization) for a vector of co-latitudes ϑ up to degree $N_{\max}$ and for all orders $0 \leq m \leq n$ via recursion formulas. In addition, the 1st and 2nd derivatives $\frac{\mathrm{d}\overline{P}_{n,m}(\cos\vartheta)}{\mathrm{d}\vartheta}$ and $\frac{\mathrm{d}^2\overline{P}_{n,m}(\cos\vartheta)}{\mathrm{d}\vartheta^2}$ w. r. t. the co-latitude ϑ are provided.
legendreP_Xnum.m	the same as *legendreP.m*, but enhanced by X-numbers for high degree/order.
legpol.m	calculates the unnormalized Legendre functions for a vector of co-latitudes ϑ for a vector of degrees n and a scalar order m.
LeNorm.m	provides order dependent factors to switch from the geodetic normalization of spherical harmonic functions to other conventions. This is in particular necessary for rotating the spherical harmonic functions via Wigner-d-functions.
lovenr.m	interpolates the Love numbers of the elastic Earth up to degree 200.
mex_compile.m	demonstrates how to compile and integrate mex-files into SHBundle (currently *Legendre_mex.m*).
nablaPot.m	determines the partial derivatives for a vector of locations (i. e. pointwise) in spherical coordinates, i. e. $\frac{\mathrm{d}V(\lambda,\vartheta,r)}{\mathrm{d}r}$, $\frac{\mathrm{d}V(\lambda,\vartheta,r)}{\mathrm{d}\lambda}$ and $\frac{\mathrm{d}V(\lambda,\vartheta,r)}{\mathrm{d}\vartheta}$.
neumann.m	returns the weights and nodes for Neumann's numerical integration on the sphere.

normalklm.m	returns an ellipsoidal normal field – consisting of normalized $\{-J_0, -J_2, -J_4, -J_6, -J_8\}$ coefficients – in the |C\S|-format.
ordrms.m	computes the order RMS of a set of spherical harmonic coefficients.
parse_icgem.m	parses coefficient files given in ICGEM-format and returns them in clm-format (2.27).
plm.m	calculates the associated Legendre functions (in geodetic normalization) for a vector of co-latitudes ϑ, for a vector of degrees n and a scalar order m. In addition, the 1st and 2nd derivatives $\frac{\mathrm{d}\overline{P}_{n,m}(\cos\vartheta)}{\mathrm{d}\vartheta}$ and $\frac{\mathrm{d}^2\overline{P}_{n,m}(\cos\vartheta)}{\mathrm{d}\vartheta^2}$ w. r. t. the co-latitude ϑ are provided.
pshs.m	determines the 1st to 4th radial derivative for the disturbing potential along the orbit, which is given by vectorial coordinates.
real2cpxcov.m	generates a $(N_{\max}+1)^2 \times (N_{\max}+1)^2$ matrix for converting a real-valued spherical harmonic spectral covariance matrix into a complex-valued one for the given degree $N_{\max}$.
real2cpxmat.m	generates a real to complex conversion of spherical harmonics for the given degree.
real2cpxsh.m	converts the real spherical harmonics to complex spherical harmonics.
rotate_shc.m	applies the rotation to the spherical harmonic coefficients instead of the functions.
regionalshs.m	enables a regional spherical harmonic synthesis but for a regular grid.
sc2cs.m	rearranges spherical harmonic coefficients from the the /S|C\-format (2.30) to |C\S|-format (2.29) (inverse of *cs2sc.m*).
shplot.m	visualizes the spherical harmonic coefficients in a "triangle plot".
shprepare.m	prepares a spherical harmonic spectrum for imaging purposes, i. e. in the /S|C\-format, non existing entries are masked, etc.
shspkamph.m	computes the amplitude and phase quantities of a given spherical harmonic spectrum.
sortLegendre.m	provides an index for re-arranging between the vector formats.
sph_gradshs.m	calculates the gradient of a potential described in spherical harmonics for a single point (or for two points) in space. The function handle on an internal function is a newer feature of MATLAB, older versions or alternatives like OCTAVE might be unable to run this code.
tenspshs.m	determines the gradient and the tensor of a field given in spherical harmonic coefficients for a vector of locations (i. e. pointwise) in the traditional expressions including a singularity at the poles.

updwcont.m	approximates the upward continuation of the disturbing potential by a Taylor series for given radial derivatives.
upwcon.m	provides the factor $(R/r)^n$ for the upward continuation of field quantities.
vec2cs.m	rearranges spherical harmonic coefficients from the vector-format (2.31) to the `|C\S|`-format (2.29) (inverse of *cs2vec.m*).
wigner_all.m	computes the Wigner-d-functions up to degree $N_{\max}$ and for all integer orders $0 \leq k, m \leq n$ via recursion formulas. The inclination functions are found by multiplication with *Legendre0.m*.
ylm.m	calculates the normalized spherical harmonics (= product of Legendre functions and exponential functions) for a scalar order and degree and a regular global grid.
ylmplot.m	visualizes the normalized spherical harmonics (= product of Legendre functions and exponential functions) for a scalar order and degree and a regular global grid.

Please note, that a shortened and sorted list of the routines can be found in the *Contents.m*-file and on the screen by command »*help SHbundle*« (if the toolbox is included in the MATLAB path before).

Bibliography

Devaraju, B. (2015), Understanding filtering on the sphere – Experiences from filtering GRACE data, PhD thesis, University of Stuttgart.

Fukushima, T. (2012), 'Numerical computation of spherical harmonics of arbitrary degree and order by extending exponent of floating point numbers', *Journal of Geodesy* **86**(4), 271–285.

Fukushima, T. (2017), 'Rectangular rotation of spherical harmonic expansion of arbitrary-high degree and order', *Journal of Geodesy* pp. 271–285.

Gooding, R. H. and Wagner, C. A. (2010), 'On a Fortran procedure for rotating spherical-harmonic coefficients', *Celestial Mechanics and Dynamical Astronomy* **108**(1), 95–106.

Haines, G. V. (1988), 'Computer programs for spherical cap harmonic analysis of potential and general fields'.

Ilk, K. H. (1983), *Ein Beitrag zur Dynamik ausgedehnter Körper: Gravitationswechselwirkung*, C: Deutsche Geodätische Kommission bei der Bayerischen Akademie der Wissenschaften, Beck.

Koop, R. (1993), Global Gravity Field Modelling Using Satellite Gravity Gradiometry, PhD thesis.

Kostelec, P. J. and Rockmore, D. N. (2003), 'FFTs on the Rotation Group'.

Mayer-Gürr, T. (2006), Gravitationsfeldbestimmung aus der Analyse kurzer Bahnbögen am Beispiel der Satellitenmissionen CHAMP und GRACE, PhD thesis, Rheinische Friedrich-Wilhelms-Universität, Bonn.

Petrovskaya, M. and Vershkov, A. (2006), 'Non-singular expressions for the gravity gradients in the local north-oriented and orbital reference frames', *Journal of Geodesy* **80**(3), 117–127.

Simons, F. J. and Dahlen, F. A. (2006), 'Spherical Slepian functions and the polar gap in geodesy', *Geophys. J. Int.* **166**, 1039–1061.

Simons, F. J., Dahlen, F. A. and Wieczorek, M. A. (1961), 'Spatiospectral Concentrationon a Sphere', *SIAM Review* **40**(1), 43–63.

Slepian, D. and Pollak, H. (1961), 'Prolate spheroidal wavefunctions, Fourier analysis and uncertainty – I', *Bell Syst Tech J* **48**(3), 504–536.

Sneeuw, N. (1991), Inclination functions: Group theoretical background and a recursive algorithm, Technical report, Delft University of Technology.

Sneeuw, N. (2012), Inclination functions: Orthogonality and other properties, *in* N. Sneeuw, P. Novák, M. Crespi and F. Sansó, eds, 'VII Hotine-Marussi Symposium on Mathematical Geodesy', Vol. 137 of *International Association of Geodesy Symposia*, Springer Berlin Heidelberg, pp. 267–272.

Wieczorek, M. A. and Simons, F. J. (2005), 'Localized spectral analysis on the sphere', *Geophys. J. Int* **162**, 655–675.

Xue, Y. and Sneeuw, N. (2006), 'Ein stabiler Algorithmus zur Berechnung von Legendre-Funktionen hohen Grades', Poster.