April 7, 2003

$SO(3)$: Definitions, Conventions, and Code

Definitions, normalizations ete etc are all taken from [1].

1. **Euler angle decomposition**: Let

$$u(A) = \begin{pmatrix} \cos A & -\sin A & 0 \\ \sin A & \cos A & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad a(B) = \begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix} \tag{1}$$

   Then $g \in SO(3)$ can be written as $g = u(\alpha)\, a(\beta)\, u(\gamma)$ where for $0 \le \alpha, \gamma < 2\pi$ and $0 \le \beta \le \pi$.

2. **Wigner d-function**:

$$\begin{aligned} d^J_{MM'}(\beta) &= \zeta_{MM'} \sqrt{\frac{s!(s+\mu+\nu)!}{(s+\mu)!(s+\nu)!}} \left(\sin\frac{\beta}{2}\right)^{\mu} \left(\cos\frac{\beta}{2}\right)^{\nu} \\ &\quad \times P_s^{(\mu,\nu)}(\cos\beta) \end{aligned} \tag{2}$$

   where

$$\mu = |M - M'| \quad \nu = |M + M'| \quad s = J - \frac{\mu+\nu}{2}$$

   and

$$\zeta_{MM'} = \begin{cases} 1 & \text{if } M' \ge M \\ (-1)^{M'-M} & \text{if } M' < M. \end{cases}$$

   and $P_s^{(\mu,\nu)}(\cos\beta)$ is a Jacobi polynomial. Note that unless $J \ge \max(|M|, |M'|)$, we have $d^J_{MM'}(\beta) = 0$.

   - **orthogonality condition**:

$$\int_0^{\pi} d^J_{MM'}(\beta) d^{J'}_{MM'}(\beta)\ \sin\beta\ d\beta = \frac{2}{2J+1}\delta_{JJ'}, \tag{3}$$

   - **three-term recurrence**:

$$\begin{aligned} 0 &= \frac{\sqrt{\left[(J+1)^2 - M^2\right]\left[(J+1)^2 - M'^2\right]}}{(J+1)(2J+1)}\ d^{J+1}_{MM'}(\beta) + \left(\frac{MM'}{J(J+1)} - \cos\beta\right) d^J_{MM'}(\beta) \\ &\quad + \frac{\sqrt{(J^2 - M^2)(J^2 - M'^2)}}{J(2J+1)}\ d^{J-1}_{MM'}(\beta) \end{aligned} \tag{4}$$

   - **C code uses normalized versions of Wigner d-functions**:

$$\tilde{d}^J_{MM'}(\beta) = \sqrt{\frac{2J+1}{2}}\ d^J_{MM'}(\beta). \tag{5}$$

   - **normalized version of three-term recurrence; used in the C code**:

$$\begin{aligned} \tilde{d}^{J+1}_{MM'}(\beta) &= \sqrt{\frac{2J+3}{2J+1}}\ \frac{(J+1)(2J+1)}{\sqrt{\left[(J+1)^2 - M^2\right]\left[(J+1)^2 - M'^2\right]}} \left(\cos\beta - \frac{MM'}{J(J+1)}\right) \tilde{d}^J_{MM'}(\beta) \\ &\quad - \sqrt{\frac{2J+3}{2J-1}}\ \frac{\sqrt{\left[J^2 - M^2\right]\left[J^2 - M'^2\right]}}{\sqrt{\left[(J+1)^2 - M^2\right]\left[(J+1)^2 - M'^2\right]}}\ \frac{J+1}{J}\ \tilde{d}^{J-1}_{MM'}(\beta). \end{aligned} \tag{6}$$

- **initialization definitions - C code uses normalized versions**

$$
\begin{aligned}
d^J_{JM}(\beta) &= \sqrt{\frac{(2J)!}{(J+M)!(J-M)!}} \left(\cos\frac{\beta}{2}\right)^{J+M} \left(-\sin\frac{\beta}{2}\right)^{J-M} \\
d^J_{-JM}(\beta) &= \sqrt{\frac{(2J)!}{(J+M)!(J-M)!}} \left(\cos\frac{\beta}{2}\right)^{J-M} \left(\sin\frac{\beta}{2}\right)^{J+M} \\
d^J_{MJ}(\beta) &= \sqrt{\frac{(2J)!}{(J+M)!(J-M)!}} \left(\cos\frac{\beta}{2}\right)^{J+M} \left(\sin\frac{\beta}{2}\right)^{J-M} \\
d^J_{M-J}(\beta) &= \sqrt{\frac{(2J)!}{(J+M)!(J-M)!}} \left(\cos\frac{\beta}{2}\right)^{J-M} \left(-\sin\frac{\beta}{2}\right)^{J+M}.
\end{aligned} \tag{7}
$$

3. **Wigner D-functions**:the collection of functions form a complete set of orthogonal functions with respect to integration over $SO(3)$:

$$
D^J_{MM'}(\alpha,\beta,\gamma) = e^{-iM\alpha}\, d^J_{MM'}(\beta)\, e^{-iM'\gamma}, \qquad J, M, M' \text{ integers} \tag{8}
$$

- **orthogonality**:

$$
\int_0^{2\pi} d\alpha \int_0^{\pi} d\beta \sin\beta \int_0^{2\pi} d\gamma \;\; D^{J_2 *}_{M_2 M'_2}(\alpha,\beta,\gamma) D^{J_1}_{M_1 M'_1}(\alpha,\beta,\gamma) = \frac{8\pi^2}{2J_1+1} \delta_{J_1 J_2}\; \delta_{M_1 M_2}\; \delta_{M'_1 M'_2}. \tag{9}
$$

The C code uses a **normalized** version of the D-functions:

$$
\tilde{D}^J_{MM'}(\alpha,\beta,\gamma) = \frac{1}{2\pi}\sqrt{\frac{2J+1}{2}}\; D^J_{MM'}(\alpha,\beta,\gamma) \tag{10}
$$

$$
\int_0^{2\pi} d\alpha \int_0^{\pi} d\beta \sin\beta \int_0^{2\pi} d\gamma \;\; \tilde{D}^{J_2 *}_{M_2 M'_2}(\alpha,\beta,\gamma) \tilde{D}^{J_1}_{M_1 M'_1}(\alpha,\beta,\gamma) = \delta_{J_1 J_2}\; \delta_{M_1 M_2}\; \delta_{M'_1 M'_2}. \tag{11}
$$

- **decomposition**: $f \in L^2(SO(3))$:

$$
f(\alpha,\beta,\gamma) = \sum_{J\geq 0} \sum_{M=-J}^{J} \sum_{M'=-J}^{J} \hat{f}^J_{MM'} D^J_{MM'}(\alpha,\beta,\gamma) \tag{12}
$$

where

$$
\begin{aligned}
\hat{f}^J_{MM'} &= \left\langle f, D^J_{MM'} \right\rangle \\
&= \frac{2J+1}{8\pi^2} \int_0^{2\pi} d\alpha \int_0^{\pi} d\beta \sin\beta \int_0^{2\pi} d\gamma \;\; f(\alpha,\beta,\gamma) D^{J*}_{MM'}(\alpha,\beta,\gamma).
\end{aligned} \tag{13}
$$

4. **Band-limit definition**: (Band-limited functions on $SO(3)$). A continuous function f on $SO(3)$ is **band-limited with band-limit (or bandwidth)** B if $\hat{f}^l_{MM'} = 0$ for all $l \geq B$.

5. **quadrature weights**:

$$
w_B(j) = \frac{2}{B} \sin\left(\frac{\pi(2j+1)}{4B}\right) \sum_{k=0}^{B-1} \frac{1}{2k+1} \sin\left((2j+1)(2k+1)\frac{\pi}{4B}\right). \tag{14}
$$

6. **Discrete Wigner-d transform**: For given integers (M, M'), define the **Discrete Wigner Transform (DWT) of a data vector s** to be the collection of sums of the form

$$\hat{\mathbf{s}}(l, M, M') = \sum_{k=0}^{2B-1} w_B(k)\ \tilde{d}^l_{M,M'}(\beta_k)[\mathbf{s}]_k \qquad \max(|M|, |M'|) \le l < B \tag{15}$$

where $\tilde{d}^l_{M,M'}$ is a Wigner d-function of degree l and orders M, M', and $\beta_k = \dfrac{\pi(2k+1)}{4B}$.

7. **Discrete $SO(3)$ transform at bandwidth B**: the discrete version of (13), but using normalized Wigner-D's:

$$f^l_{MM'} = \frac{1}{(2B)^2} \sum_{j_1=0}^{2B-1} \sum_{j_2=0}^{2B-1} \sum_{k=0}^{2B-1} w_B(k) f(\alpha_{j_1}, \beta_k, \gamma_{j_2}) \tilde{D}^{l*}_{MM'}(\alpha_{j_1}, \beta_k, \gamma_{j_2}) \tag{16}$$

where the function is sampled on the $2B \times 2B \times 2B$ grid $\alpha_{j_1} = \dfrac{2\pi j_1}{2B}$, $\beta_k = \dfrac{\pi(2k+1)}{4B}$, $\gamma_{j_2} = \dfrac{2\pi j_2}{2B}$.

8. **C code user routines, in no particular order**:
   (a) `test_FST_SO3`: does an inverse-forward $SO(3)$ transforms, to test code stability
   (b) `test_FST_SO3_For`: does a normalized forward $SO(3)$ transform, that is, computes (16) for all legal l, M, M'; needs input data from the user
   (c) `test_FST_SO3_Inv`: does a normalized inverse $SO(3)$ transform; needs input data from the user
   (d) `test_Wigner_Analysis`: does a normalized forward Discrete Wigner-d transform, i.e. eqn. (15); needs user input
   (e) `test_Wigner_Synthesis`: does a normalized inverse Discrete Wigner-d transform; needs user input
   (f) `test_genWig`: routine to generate all the normalized Wigner-d functions, eqn. (5), at a particular bandwidth B, and orders M, M'.
   (g) `test_wigSpec`: for a particular bandwidth B, and orders M, M', generates the appropriate normalized Wigner-d function (to start the recurrence), i.e. eqn. (7).

9. **Data files provided with source code**: Included are samples of functions, at various orders, degrees and bandwidths. The files are in real-imaginary pairs, e.g. `D101real_bw4.dat`, `D101imag_bw4.dat` contain the real and imaginary parts, resp., of the normalized Wigner-D function $\tilde{D}^1_{01}$ (i.e. $J = 1, M = 0, M' = 1$) sampled on appropriate bandwidth $= 4$ grid (which has dimension $8 \times 8 \times 8$, so there are 512 values in each of these files).

   Filenames are of the form `DJMM'real_bw?.dat`, `DJMM'imag_bw?.dat`, so, as another example, the files `D3-11real_bw4.dat`, `D3-11imag_bw4.dat` contain the 8^3 values of $\tilde{D}^3_{-11}$ (i.e. $J = 3, M = -1, M' = 1$) sampled on the proper bandwidth $= 4$ grid.

   Note: one exception to this rule. The files

   `D101real_bw4.dat`, `D101imag_bw4.dat`

   actually contain samples of $(2 + \imath)D^1_{01}$. I wanted to get something slightly non-trivial in there.

   One more set of data files. These occur in groups of four:

   `rand1R_bwX.dat rand1I_bwX.dat rand1Rcoeff_bwX.dat rand1Icoeff_bwX.dat`

   contain the real and imaginary parts of sample values of a random (and bandlimited) function on $SO(3)$, as the real and imaginary parts of its coefficients.

10. **file formats: samples and coefficients**: For all that follows, we're dealing with a fixed bandwidth B. First, let's deal with the samples. Recall from above that $\alpha_{j_1} = \dfrac{2\pi j_1}{2B}$, $\beta_k = \dfrac{\pi(2k+1)}{4B}$, $\gamma_{j_2} = \dfrac{2\pi j_2}{2B}$, where $0 \le k, j_1, j_2 \le 2B - 1$. This is where the function is sampled for a bandwidth B transform. The C

code expects the two files, containing the real and imaginary parts of the sample values, to be ordered as follows:

$$\begin{array}{r} f(\alpha_0, \beta_0, \gamma_0) \\ f(\alpha_0, \beta_0, \gamma_1) \\ ... \\ f(\alpha_0, \beta_0, \gamma_{2B-1}) \\ f(\alpha_1, \beta_0, \gamma_0) \\ f(\alpha_1, \beta_0, \gamma_1) \\ ... \\ f(\alpha_{2B-1}, \beta_0, \gamma_{2B-1}) \\ f(\alpha_0, \beta_1, \gamma_0) \\ f(\alpha_0, \beta_1, \gamma_1) \\ ... \\ f(\alpha_{2B-1}, \beta_{2B-1}, \gamma_{2B-1}) \end{array}$$

So, of the three indeces, j_2 iterates the fastest, and k the slowest. Think of it as sampling at all legal longitudes for each latitude. That's how the S^2 transform works.

Now for the coefficients. As with the sample values, there will be two files, one each for real and imaginary parts, and there will be one number per line. As for the rest, it might seem a little weird, but bear with me. Recall how the spherical coefficients are indexed from (16): $\hat{f}^l_{MM'}$.

Consider a matrix A whose **rows** are indexed by M as follows:

$$M = 0, 1, 2, ..., B-1, -(B-1), -(B-2), ..., -1$$

This is the order they occur, e.g. if $B = 4$, then the fifth row corresponds to $M = -3$. Similarly for the columns, indexed by M':

$$M' = 0, 1, 2, ..., B-1, -(B-1), -(B-2), ..., -1$$

E.g. the seventh column corresponds to $M' = -1$. Ok, now I reveal that the element at $A(i, j)$ is actually an **array** which contains the $SO(3)$ coefficients

$$\{f^l_{ij} = < f, \tilde{D}^l_{ij} > \ \mid \ \max(|i|, |j|) \leq l \leq B-1\}$$

Now, *finally*, write down this matrix A in row-major format. E.g. First write down the set of coefficients for $M = 0, M' = 0$, then for $M = 0, M' = 1$, then for ... , then for $M = 0, M' = -(B-1)$, then for ..., then for $M = 0, M' = -1$, then for $M = 1, M' = 0$, and then $M = 1, M' = 1$, and so on. You get the idea. Believe me, in some sense, this is natural.

To make things easier, here are four formulae which will tell you where in the list the $SO(3)$ coefficient $f^l_{MM'}$ occurs. There are four functions, depending on the signs of M, M'. These formulae can be simplified, but then they might seem a little more mysterious.

Let B denote the bandwidth, $h(M, M', B) = B - \max(|M|, |M'|)$. Then the location of $f^l_{MM'}$ in the file is

$$\sum_{k=0}^{M-1}(B^2 - k^2) + \sum_{k=0}^{M'-1} h(M, k, B) + (l - \max(M, M')) + 1 \qquad \text{if } M, M' \geq 0 \tag{17}$$

$$\tag{18}$$

$$\sum_{k=0}^{M}(B^2 - k^2) - \sum_{k=M'}^{-1} h(M, k, B) + (l - \max(M, |M'|)) + 1 \quad \text{if } M \geq 0,\ M' < 0 \tag{19}$$

$$\tag{20}$$

$$\frac{4B^3 - B}{3} - \sum_{k=1}^{|M|}(B^2 - k^2) + \sum_{k=0}^{M'-1} h(M, k, B) + (l - \max(|M|, M')) + 1 \quad \text{if } M < 0,\ M' \geq 0 \tag{21}$$

$$\tag{22}$$

$$\frac{4B^3 - B}{3} - \sum_{k=1}^{|M|-1}(B^2 - k^2) - \sum_{k=M'}^{-1} h(M, k, B) + (l - \max(|M|, |M'|)) + 1 \qquad \text{if } M, M' < 0 \tag{23}$$

If you program this in C, you don't have to do that "+1". I.e. as it's written now, the formula for $M = M' = 0$ will tell you that the location of f^0_{00} is 1.

References

[1] D. A. Varshalovich, A. N. Moskalev and V. K. Khersonskii, *Quantum Theory of Angular Momentum*, World Scientific Publishing, Singapore, 1988.