

This document outlines the response function library (rfl), developed to utilize the PRBEM Response Format to perform common calculations, such as the energy and pitch-angle response of a sensor with a given orientation.

#### Version History:

v1.0.0 – First “official” draft [TPO]

v1.1.0 – Changed XCAL to CROSSCALIB [ TPO, 8 June 2011]

The instrument coordinate axes are a right-handed system,  $s_0, s_1, s_2$ , with  $s_0$  parallel to particles traveling into the instrument boresight. A polar angle, THETA ( $\theta$ ) measures declination from  $s_0$ . An azimuth or longitude angle PHI ( $\phi$ ) measures the angle in the  $s_1$ - $s_2$  plane from  $s_1$  to  $s_2$ . Rotations between instrument coordinates and magnetic field coordinates (pitch angle  $\alpha$  and gyrophase  $\beta$ ) will be performed with attitude information provided in the Level 1 files. Polar angles  $\alpha, \theta$  range from 0 to 180°, while  $\phi$  and  $\beta$  range from 0 to 360°. The instrument coordinates are given in the Level 1 file as S0\_LAT, S0\_LON, S1\_LAT, S1\_LON, S2\_LAT, S2\_LON.

These quantities define the response function  $R$  below:

$$y \approx \lambda = \frac{1}{X_{cal}} \int_0^\infty \int_0^{2\pi} \int_0^\pi R(E, \theta, \phi) j(E, \alpha(\theta, \phi, t), \beta(\theta, \phi, t)) dt \sin \theta d\theta d\phi dE$$

When energy and angular response are separable, we have:

$$y \approx \lambda = \frac{1}{X_{cal}} \int_0^\infty \varepsilon(E) \int_0^{2\pi} \int_0^\pi A_{eff}(\theta, \phi) j(E, \alpha(\theta, \phi, t), \beta(\theta, \phi, t)) dt \sin \theta d\theta d\phi dE$$

We usually wish to compute weights  $h_{ijk}$ , such that the integral can be replaced by a sum:

$$y \approx \lambda \approx \sum_{ijk} h_{ijk} j(E_i, \alpha_j, \beta_k) = \sum_{ijk} h_{ijk} j_{ijk}$$

This is particularly helpful for inversion and data assimilation.

Manipulations to define  $h_{ijk}$  explicitly:

$$\begin{aligned} y \approx \lambda &= \frac{1}{X_{cal}} \int_0^\infty \int_0^{2\pi} \int_0^\pi j(E, \alpha, \beta) \left[ \int_{t_1}^{t_2} R(E, \theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt \right] d\beta d\alpha dE \\ &\approx \frac{1}{X_{cal}} \sum_{ijk} j_{ijk} \left[ \int_{t_1}^{t_2} R(E_i, \theta(\alpha_j, \beta_k, t), \phi(\alpha_j, \beta_k, t)) dt \right] \Delta\beta_k \Delta\cos\alpha_j \Delta E_i \\ h_{ijk} &= \frac{1}{X_{cal}} \left[ \int_{t_1}^{t_2} R(E_i, \theta(\alpha_j, \beta_k, t), \phi(\alpha_j, \beta_k, t)) dt \right] \Delta\beta_k \Delta\cos\alpha_j \Delta E_i \end{aligned}$$

When we assume gyrotropy, i.e.,  $j(E, \alpha)$ , we have:

$$h_{ij} = \frac{1}{X_{cal}} \left[ \int_0^{2\pi t_2} \int_0^{t_1} R(E_i, \theta(\alpha_j, \beta, t), \phi(\alpha_j, \beta, t)) dt d\beta \right] \Delta \cos \alpha_j \Delta E_i$$

When we also assume bounce symmetry, i.e.,  $j(E, \alpha) = j(E, \pi - \alpha)$ , our  $\alpha$  grid is limited to  $[0, \pi/2]$ , and we have:

$$h_{ij} = \frac{1}{X_{cal}} \left[ \int_0^{2\pi t_2} \int_0^{t_1} [R(E_i, \theta(\alpha_j, \beta, t), \phi(\alpha_j, \beta, t)) + R(E_i, \theta(\pi - \alpha_j, \beta, t), \phi(\pi - \alpha_j, \beta, t))] dt d\beta \right] \Delta \cos \alpha_j \Delta E_i$$

We will need conversions between magnetic angles  $(\alpha, \beta)$  and instrument angles  $(\theta, \phi)$ . Basis vectors for the magnetic coordinate systems are  $(c, d, b)$ , where  $b$  is the magnetic field direction,  $c$  defines the direction for  $\beta=0$ , and  $d$  completes the right-handed system. Basis vectors for the instrument coordinates are  $(s_1, s_2, s_0)$ , defined above. Arbitrary vectors in the two frames are given by:

$$\hat{s} = \hat{b} \cos \alpha + \hat{c} \sin \alpha \cos \beta + \hat{d} \sin \alpha \sin \beta$$

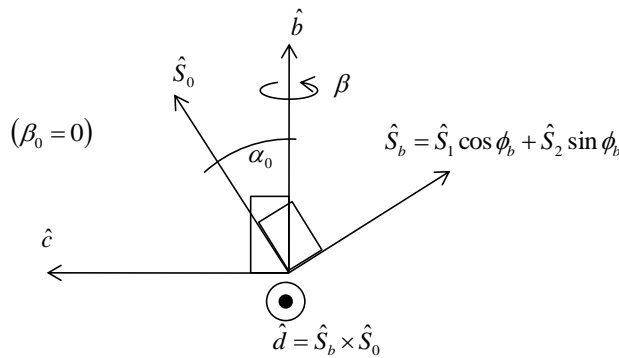
$$\hat{s} = \hat{s}_0 \cos \theta + \hat{s}_1 \sin \theta \cos \phi + \hat{s}_2 \sin \theta \sin \phi$$

The normal incidence direction  $s_0$  has pitch angle  $\alpha_0$  and gyrophase  $\beta_0$ , while the magnetic field has colatitudes  $\alpha_0$  and longitude  $\phi_b$  in the instrument frame:

$$\hat{s}_0 = \hat{b} \cos \alpha_0 + \hat{c} \sin \alpha_0 \cos \beta_0 + \hat{d} \sin \alpha_0 \sin \beta_0$$

$$\hat{b} = \hat{s}_0 \cos \alpha_0 + \hat{s}_1 \sin \alpha_0 \cos \phi_b + \hat{s}_2 \sin \alpha_0 \sin \phi_b$$

The figure below illustrates the case for  $\beta_0=0$ . Note that for  $\alpha_0=0$ ,  $\beta_0=0$ ,  $\phi_b=0$ , the  $s_0$  axis is parallel to the  $b$  axis, the  $s_1$  axis is anti-parallel to the  $c$  axis, and the  $s_2$  axis is anti-parallel to the  $d$  axis. This anti-parallel topology arises from the choice of having  $s_0$  in the first octant of the  $c$ - $d$ - $b$  coordinates, and  $b$  in the first octant of the  $s_1$ - $s_2$ - $s_0$  coordinates. Also note that  $b$ - $c$ - $d$  coordinates are equivalent to  $c$ - $d$ - $b$  coordinates, and  $s_0$ - $s_1$ - $s_2$  coordinates are equivalent to  $s_1$ - $s_2$ - $s_0$  coordinates, but we have followed the convention of having the polar axis as the third coordinate.



The two coordinate systems are related to each other by a set of Euler rotations: First, a rotation of  $\pi - \beta_0$  around the  $s_0$  axis, the  $\alpha_0$  around the  $s_1$  axis, and finally  $\phi_b$  around the  $s_0$  axis.

$$\underline{\underline{\mathfrak{R}}} = \begin{bmatrix} -\sin \beta_0 \sin \phi_b - \cos \alpha_0 \cos \beta_0 \cos \phi_b & \cos \phi_b \sin \beta_0 - \cos \alpha_0 \cos \beta_0 \sin \phi_b & \sin \alpha_0 \cos \beta_0 \\ \cos \beta_0 \sin \phi_b - \cos \alpha_0 \sin \beta_0 \cos \phi_b & -\cos \beta_0 \cos \phi_b - \cos \alpha_0 \sin \beta_0 \sin \phi_b & \sin \alpha_0 \sin \beta_0 \\ \sin \alpha_0 \cos \phi_b & \sin \alpha_0 \sin \phi_b & \cos \alpha_0 \end{bmatrix}$$

The matrix  $\underline{\underline{\mathfrak{R}}}$  provides a rotation from the instrument basis to the magnetic basis, while  $\underline{\underline{\mathfrak{R}}}^T$  does the reverse. The coordinate frames are aligned, and  $\underline{\underline{\mathfrak{R}}}$  is the identity matrix, when  $\alpha_0=0$ ,  $\beta_0=\pi$ ,  $\phi_b=0$ .

The transform from  $\alpha, \beta$  to  $\theta, \phi$  coordinates is:

$$\begin{aligned} \cos \theta &= \mathfrak{R}_{33} \cos \alpha + \mathfrak{R}_{13} \sin \alpha \cos \beta + \mathfrak{R}_{23} \sin \alpha \sin \beta \\ &= \cos \alpha_0 \cos \alpha + \sin \alpha_0 \cos \beta_0 \sin \alpha \cos \beta + \sin \alpha_0 \sin \beta_0 \sin \alpha \sin \beta \\ \tan \phi &= \frac{\mathfrak{R}_{32} \cos \alpha + \mathfrak{R}_{12} \sin \alpha \cos \beta + \mathfrak{R}_{22} \sin \alpha \sin \beta}{\mathfrak{R}_{31} \cos \alpha + \mathfrak{R}_{11} \sin \alpha \cos \beta + \mathfrak{R}_{21} \sin \alpha \sin \beta} \end{aligned}$$

The transform from  $\theta, \phi$  to  $\alpha, \beta$  coordinates is:

$$\begin{aligned} \cos \alpha &= \mathfrak{R}_{33} \cos \theta + \mathfrak{R}_{31} \sin \theta \cos \phi + \mathfrak{R}_{32} \sin \theta \sin \phi \\ &= \cos \alpha_0 \cos \theta + \sin \alpha_0 \cos \phi_b \sin \theta \cos \beta + \sin \alpha_0 \sin \phi_b \sin \theta \sin \beta \\ \tan \beta &= \frac{\mathfrak{R}_{23} \cos \theta + \mathfrak{R}_{21} \sin \theta \cos \phi + \mathfrak{R}_{22} \sin \theta \sin \phi}{\mathfrak{R}_{13} \cos \theta + \mathfrak{R}_{11} \sin \theta \cos \phi + \mathfrak{R}_{12} \sin \theta \sin \phi} \end{aligned}$$

### Implementation notes:

1. The angles  $\phi$  and  $\beta$  should be computed with a 4-quadrant arctangent function (“atan2” in most languages).
2. Where  $R$  contains any delta functions or discontinuities, analytical formulas for integration/interpolation to  $h_{ijk}$  must be used to ensure that  $\Sigma h_{ijk}$  preserves the implied (or supplied) energy-geometric factor.
3.  $X_{cal}$  is provided by CROSSCALIB in the response file (was named XCAL prior to v1.1.0 of the format specification and the RFL)

## List of Functions

The routines below loosely define the functional interface for the response function library. The routines are structured wherever possible to support implementation with object-oriented programming. The library provides routines to populate the numerical integrations weights for Cartesian grids in energy and angle. The angle can be specified either in instrument coordinates (theta, phi) or magnetic coordinates (alpha, beta). The two coordinate frames are related to each other by the angles alpha0, phib, which can be computed given the first two instrument axes and the magnetic field vector in a common coordinate system. The weights returned can be used in a single, double, or triple sum, depending on the assumed geometry. The following hierarchy shows how to compose weights for 3-D, 2-D, or 1-D integrals. (Whenever possible, the sums below should be replaced by integrals to allow for analytical expressions and high-precision integrals where desired).

$$\begin{array}{ll}
 \left\{ \begin{array}{l} h_{ijk} = \text{hEthetaphi}(i, j, k) \\ h_{ijk} = \text{hE}(i)\text{hAthetaphi}(j, k) \end{array} \right. & \left\{ \begin{array}{l} h_i = \sum_j h_{ij} \\ h_i = \text{hEiso}(i) \end{array} \right. \\
 \left\{ \begin{array}{l} h_{ij} = \sum_k h_{ijk} \\ h_{ij} = \text{hEtheta}(i, j) \\ h_{ij} = \text{hE}(i)\text{hAtheta}(j) \end{array} \right. & \left\{ \begin{array}{l} h_{jk} = \sum_i h_{ijk} \\ h_{jk} = \text{hthetaphi}(j, k) \end{array} \right. \\
 \left\{ \begin{array}{l} h_i = \sum_j h_{ij} \\ h_i = \text{hE}(i)\text{hA0} \end{array} \right. & \left\{ \begin{array}{l} h_j = \sum_k h_{jk} \\ h_j = \text{htheta}(j) \end{array} \right. \\
 \left\{ \begin{array}{l} h_{ijk} = \text{hEalphabeta}(i, j, k) \\ h_{ijk} = \text{hE}(i)\text{hAalphabeta}(j, k) \end{array} \right. & h_{ik} = \text{halphabeta}(j, k) \\
 \left\{ \begin{array}{l} h_{ij} = \sum_k h_{ijk} \\ h_{ij} = \text{hEalpha}(i, j) \\ h_{ij} = \text{hE}(i)\text{hAalpha}(j) \end{array} \right. & \left\{ \begin{array}{l} h_i = \sum_k h_{ik} \\ h_j = \text{halpha}(j) \end{array} \right.
 \end{array}$$

Note that for an integral energy channel, the integral over the energy grid is replaced by a fixed energy bandwidth of 1 to avoid the infinity.

Options allows user to specify numerical integration method: e.g., trapezoidal rule. (Note: if using trapezoidal rule beta and phi grids should include both 0 and 360 if integrating over all beta or phi, and time list should include both endpoints). The syntax below should be adapted to each language in which it is implemented.

- `[inst_info,result_code] = load_inst_info(FileName,FileType);` Populates an `inst_info` object from file `FileName`. Supported file types are `.cdf`, `.hd5`, `.xml` (and native, e.g., `.mat`)

- `[alpha0,beta0,phib,result_code] = rfl_vectors_to_euler_angles(B,C,S0,S1)`; Compute pitch angle ( $\alpha_0$ ) and gyrophase angle ( $\beta_0$ ) of boresight and longitude of B ( $\phi_b$ ), the Euler angles of the Euler rotations for converting between instrument coordinates and magnetic coordinates. All four inputs are 3-vectors in some common Cartesian coordinate system (e.g., Cartesian GEI). B is parallel to the magnetic field, C defines the B-C plane, in which  $\beta_0=0$ . **S0 points *into* the instrument, parallel to normally incident particles**. S1 defines the S0-S1 plane in which  $\phi=0$ . Vectors need not have unit length, and neither B-C nor S0-S1 need to be orthogonal.
- `[alpha0,beta0,phib,result_code] = rfl_latlon_to_euler_angles(Blat,Blon,Clat,Clon,S0lat,S0lon,S1lat,S1lon)`; Same as `vectors_to_euler_angles`, except B,C, S0, S1 specified by angle pairs. All three angle pairs are degrees latitude and longitude in some common spherical coordinate system (e.g., spherical GEO).
- `[theta,phi,result_code] = alphabeta2thetaphi(alpha,beta,alpha0,beta0,phib)`; Convert pitch angle and gyrophase to instrument angles. When  $\theta=0$ ,  $\phi=0$ .
- `[alpha,beta,result_code] = thetaphi2alphabeta(theta,phi,alpha0,beta0,phib)`; Convert instrument angles to pitch angle and gyrophase. When  $\alpha=0$ ,  $\beta=0$

*In a class/object model, the following functions are the required methods for each class*

- `[R,result_code] = R(E,theta,phi)`; compute the instrument response as a function of E, theta, and phi.
- `[hEalphabeta,result_code] = make_hEalphabeta(inst_info,Egrid,alphagrid,betagrid,tgrid,alpha0,beta0,phib,options)`; Compute weights for triple numerical integral over E, alpha, beta, assuming no symmetries
- `[hEalpha,result_code] = make_hEalpha(inst_info,Egrid,alphagrid,tgrid,alpha0,options)`; Compute weights for double numerical integral over E, alpha, assuming gyrotropy
- `[hE,result_code] = make_hEiso(inst_info,Egrid,tgrid,options)`; Compute weights for numerical integral over energy, assuming isotropy
- `[halphabeta,result_code] = make_halphabeta(inst_info,alphagrid,betagrid,tgrid,alpha0,beta0,phib,options)`; Compute weights for double numerical integral alpha, beta, assuming no symmetries, integrated over energy<sup>1</sup>.
- `[halpha,result_code] = make_halpha(inst_info, alphagrid,tgrid,alpha0,options)`; Compute weights for numerical integral over alpha, assuming gyrotropy, integrated over energy<sup>1</sup>.
- `[hEthetaphi,result_code] = make_hEthetaphi(inst_info,Egrid,thetagrid,phigrid,options)`; Compute weights for triple numerical integral over E, theta, phi, assuming no symmetries. Does not account for time integral
- `[hEtheta,result_code] = make_hEtheta(inst_info,Egrid,thetagrid,options)`; Compute weights for double numerical integral over E, theta, assuming a symmetric instrument. Does not account for time integral
- `[hEtheta,result_code] = make_hE (inst_info,Egrid,options)`; Compute weights for numerical integral over E, assuming an omnidirectional instrument. Does not account for time integral
- `[hthetaphi,result_code] = make_hthetaphi(inst_info,thetagrid,phigrid,options)`; Compute weights for double numerical integral over E, theta, phi, assuming no symmetries, integrated over energy<sup>1</sup>. Does not account for time integral.

- `[htheta,result_code] = make_htheta(inst_info, thetagrid,options)`; Compute weights for numerical integral over theta, assuming a symmetric instrument, integrated over energy<sup>1</sup>. Does not account for time integral
- `H = rfl_interp_weights_1d(xgrid,xhat,xbc)`; returns matrix of weights that left multiplies the flux vector on xgrid and results in a vector of fluxes at points xhat. The final optional argument, xbc, specifies the boundary conditions for extrapolation (clip at zero on left or right end, extrapolate on left or right end, and periodic with specified period).
- `H = rfl_interp_weights_2d(xgrid,ygrid,xhat,yhat,xbc,ybc)`; returns matrix of weights that left multiplies the flux vector (unwrapped from 2D to 1D) on xgrid-ygrid and results in a vector of fluxes at points (xhat,yhat). The optional arguments, xbc, ybc, specify the boundary conditions for extrapolation, as for `rfl_interp_weights_1d`.
- `H = rfl_interp_weights_3d(xgrid,ygrid,zgrid,xhat,yhat,zhat,xbc,ybc,zbc)`; returns matrix of weights that left multiplies the flux vector (unwrapped from 3D to 1D) on xgrid-ygrid-zgrid and results in a vector of fluxes at points (xhat,yhat,zhat). The optional arguments, xbc, ybc, zbc specify the boundary conditions for extrapolation, as for `rfl_interp_weights_1d`.

<sup>1</sup>Note: Energy integral over an “integral energy” channel is infinite, so the energy bandwidth is set to 1 to avoid the infinity.

Several examples in the Matlab implementation are provided:

- Building the response file CDF for ICO omnidirectional electron channels (`rfl_make_ico.m`)
- Build the response file CDF for SAMPEX/PET wide-angle proton channels (`rfl_make_sampex_pt.m`)
- Building and doing a little testing on all response types, makes response file CDF `all_types.cdf` (`rfl_make_all_types.m`)
- Building weights to project a global 3D flux model through the local instrument response (`rfl_resp_to_3D_model_example.m`)
- Using `rfl` to build weights for use with the inversion library (`rfl_invlib_example.m`):
  - Analytical spectral inversion with ICO electron channels
  - Angular wide2uni angular conversion using PET
  - Determining the local angular distribution from simulated spinning PET by angular principal component inversion

The table below provides the main expressions needed to compute weights for the various integrals using a trapezoidal rule (although most expressions are invariant to the choice of numerical integration technique).

Sensor Type	Integral for $\lambda$	Weights
<b>Energy-Angle Codependent</b>		
Arbitrary geometry	$\frac{1}{X_{cal}} \int_0^\infty \int_{-1}^1 \int_0^{2\pi} j(E, \alpha, \beta) \left[ \int_{t_1}^{t_2} R(E, \theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt \right] d\beta d \cos \alpha dE$	$h_{ijk} = \frac{\Delta \beta_k \Delta \cos \alpha_j \Delta E_i}{X_{cal}} \int_{t_1}^{t_2} R(E_i, \theta(\alpha_j, \beta_k, t), \phi(\alpha_j, \beta_k, t)) dt$
	$\frac{1}{X_{cal}} \int_0^\infty \int_{-1}^1 j(E, \alpha) \left[ \int_0^{2\pi} \int_{t_1}^{t_2} R(E, \theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt d\beta \right] d \cos \alpha dE$	$h_{ij} = \frac{\Delta \cos \alpha_j \Delta E_i}{X_{cal}} \int_0^{2\pi} \int_{t_1}^{t_2} R(E_i, \theta(\alpha_j, \beta, t), \phi(\alpha_j, \beta, t)) dt d\beta$
	$\frac{1}{X_{cal}} \int_0^\infty \int_0^1 j(E, \alpha) \left[ \int_0^{2\pi} \int_{t_1}^{t_2} \left( R(E, \theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) + R(E, \theta(\pi - \alpha, \beta, t), \phi(\pi - \alpha, \beta, t)) \right) dt d\beta \right] d \cos \alpha dE$	$h_{ij} = \frac{\Delta \cos \alpha_j \Delta E_i}{X_{cal}} \int_0^{2\pi} \int_{t_1}^{t_2} \left( R(E_i, \theta(\alpha_j, \beta, t), \phi(\alpha_j, \beta, t)) + R(E_i, \theta(\pi - \alpha_j, \beta, t), \phi(\pi - \alpha_j, \beta, t)) \right) dt d\beta$
	$\frac{1}{X_{cal}} \int_0^\infty j(E) \left[ \int_{-1}^1 \int_0^{2\pi} \int_{t_1}^{t_2} R(E, \theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt d\beta d \cos \alpha \right] dE$	$h_i = \frac{\Delta E_i (t_2 - t_1)}{X_{cal}} \int_{-1}^1 \int_0^{2\pi} R(E_i, \theta, \phi) d\phi d \cos \theta$
Cylindrically symmetric	$\frac{1}{X_{cal}} \int_0^\infty \int_{-1}^1 \int_0^{2\pi} j(E, \alpha, \beta) \int_{t_1}^{t_2} R(E, \theta(\alpha, \beta, t)) dt d\beta d \cos \alpha dE$	$h_{ijk} = \frac{\Delta \beta_k \Delta \cos \alpha_j \Delta E_i}{X_{cal}} \int_{t_1}^{t_2} R(E_i, \theta(\alpha_j, \beta_k, t)) dt$
	$\frac{1}{X_{cal}} \int_0^\infty \int_{-1}^1 j(E, \alpha) \int_0^{2\pi} \int_{t_1}^{t_2} R(E, \theta(\alpha, \beta, t)) dt d\beta d \cos \alpha dE$	$h_{ij} = \frac{\Delta \cos \alpha_j \Delta E_i}{X_{cal}} \int_{t_1}^{t_2} \int_0^{2\pi} R(E_i, \theta(\alpha_j, \beta, t)) d\beta dt \text{ (see note 1)}$
	$\frac{1}{X_{cal}} \int_0^\infty \int_0^1 j(E, \alpha) \int_0^{2\pi} \int_{t_1}^{t_2} \left( R(E, \theta(\alpha, \beta, t)) + R(E, \theta(\pi - \alpha, \beta, t)) \right) dt d\beta d \cos \alpha dE$	$h_{ij} = \frac{\Delta \cos \alpha_j \Delta E_i}{X_{cal}} \int_{t_1}^{t_2} \int_0^{2\pi} \left( R(E_i, \theta(\alpha_j, \beta, t)) + R(E_i, \theta(\pi - \alpha_j, \beta, t)) \right) d\beta dt \text{ (see note 1)}$
	$\frac{1}{X_{cal}} \int_0^\infty j(E) \int_{-1}^1 \int_0^{2\pi} \int_{t_1}^{t_2} R(E, \theta(\alpha, \beta, t)) dt d\beta d \cos \alpha dE$	$h_i = \frac{\Delta E_i (t_2 - t_1)}{X_{cal}} \int_{-1}^1 \int_0^{2\pi} R(E_i, \theta) d\phi d \cos \theta$
<b>Angular Response (see notes)</b>		
Arbitrary geometry	$\frac{1}{X_{cal}} \int_0^\infty \int_{-1}^1 \int_0^{2\pi} j(E, \alpha, \beta) \left[ \int_{t_1}^{t_2} A(\theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt \right] d\beta d \cos \alpha \varepsilon(E) dE$	$h_{jk}^{(A)} = \Delta \beta_k \Delta \cos \alpha_j \int_{t_1}^{t_2} A(\theta(\alpha_j, \beta_k, t), \phi(\alpha_j, \beta_k, t)) dt$
	$\frac{1}{X_{cal}} \int_0^\infty \int_{-1}^1 j(E, \alpha) \left[ \int_0^{2\pi} \int_{t_1}^{t_2} A(\theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt d\beta \right] d \cos \alpha \varepsilon(E) dE$	$h_j^{(A)} = \Delta \cos \alpha_j \int_0^{2\pi} \int_{t_1}^{t_2} A(\theta(\alpha_j, \beta, t), \phi(\alpha_j, \beta, t)) dt d\beta$

Sensor Type	Integral for $\lambda$	Weights
	$\frac{1}{X_{cal}} \int_0^\infty \int_0^1 j(E, \alpha) \left[ \int_0^{2\pi t_2} \left( A(\theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) + A(\theta(\pi - \alpha, \beta, t), \phi(\pi - \alpha, \beta, t)) \right) dt d\beta \right] d \cos \alpha \varepsilon(E) dE$	$h_j^{(A)} = \Delta \cos \alpha_j \int_0^{2\pi t_2} \left( A(\theta(\alpha_j, \beta, t), \phi(\alpha_j, \beta_k, t)) + A(\theta(\pi - \alpha_j, \beta, t), \phi(\pi - \alpha_j, \beta_k, t)) \right) dt d\beta$
	$\frac{1}{X_{cal}} \int_0^\infty j(E) \left[ \int_{-1}^1 \int_0^{2\pi t_2} A(\theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt d\beta d \cos \alpha \right] \varepsilon(E) dE$	$h^{(A)} = (t_2 - t_1) \int_{-1}^1 \int_0^{2\pi} A(\theta, \phi) d\phi d \cos \theta$
Cylindrically symmetric	$\frac{1}{X_{cal}} \int_0^\infty \int_{-1}^1 \int_0^{2\pi} j(E, \alpha, \beta) \left[ \int_{t_1}^{t_2} A(\theta(\alpha, \beta, t)) dt \right] d\beta d \cos \alpha \varepsilon(E) dE$	$h_{jk}^{(A)} = \Delta \beta_k \Delta \cos \alpha_j \int_{t_1}^{t_2} A(\theta(\alpha_j, \beta_k, t)) dt$
	$\frac{1}{X_{cal}} \int_0^\infty \int_{-1}^1 j(E, \alpha) \left[ \int_0^{2\pi t_2} A(\theta(\alpha, \beta, t)) dt d\beta \right] d \cos \alpha \varepsilon(E) dE$	$h_j^{(A)} = \Delta \cos \alpha_j \int_0^{2\pi t_2} A(\theta(\alpha_j, \beta, t)) dt d\beta \text{ (see note 1)}$
	$\frac{1}{X_{cal}} \int_0^\infty \int_0^1 j(E, \alpha) \left[ \int_0^{2\pi t_2} \left( A(\theta(\alpha, \beta, t)) + A(\theta(\pi - \alpha, \beta, t)) \right) dt d\beta \right] d \cos \alpha \varepsilon(E) dE$	$h_j^{(A)} = \Delta \cos \alpha_j \int_0^{2\pi t_2} \left( A(\theta(\alpha_j, \beta, t)) + A(\theta(\pi - \alpha_j, \beta, t)) \right) dt d\beta \text{ (see note 1)}$
	$\frac{1}{X_{cal}} \int_0^\infty j(E) \left[ \int_{-1}^1 \int_0^{2\pi t_2} A(\theta(\alpha, \beta, t)) dt d\beta d \cos \alpha \right] \varepsilon(E) dE$	$h^{(A)} = 2\pi(t_2 - t_1) \int_{-1}^1 A(\theta) d \cos \theta$
Omni, bidirectional (true omni)	$\frac{1}{X_{cal}} \frac{G(t_2 - t_1)}{4\pi} \int_0^\infty \int_{-1}^1 \int_0^{2\pi} j(E, \alpha, \beta) d\beta d \cos \alpha \varepsilon(E) dE$	$h_{jk}^{(A)} = \frac{G(t_2 - t_1)}{4\pi} \Delta \beta_k \Delta \cos \alpha_j$
	$\frac{1}{X_{cal}} \frac{G(t_2 - t_1)}{2} \int_0^\infty \int_{-1}^1 j(E, \alpha) d \cos \alpha \varepsilon(E) dE$	$h_j^{(A)} = \frac{G(t_2 - t_1)}{2} \Delta \cos \alpha_j$
	$\frac{1}{X_{cal}} G(t_2 - t_1) \int_0^\infty \int_0^1 j(E, \alpha) d \cos \alpha \varepsilon(E) dE$	$h_j^{(A)} = G(t_2 - t_1) \Delta \cos \alpha_j$
	$\frac{1}{X_{cal}} G(t_2 - t_1) \int_0^\infty j(E) \varepsilon(E) dE$	$h^{(A)} = G(t_2 - t_1)$



Sensor Type	Integral for $\lambda$	Weights
Omni, non-bidirectional (half omni)	$\frac{1}{X_{cal}} \frac{G}{2\pi} \int_0^\infty \int_{-1}^1 \int_0^{2\pi} j(E, \alpha, \beta) \int_{t_1}^{t_2} H(\cos \theta(\alpha, \beta, t)) dt d\beta d \cos \alpha \varepsilon(E) dE$	$h_{jk}^{(A)} = \frac{G\Delta\beta_k\Delta\cos\alpha_j}{2\pi} \int_{t_1}^{t_2} H(\cos \theta(\alpha_j, \beta_k, t)) dt \quad (\text{see note 3})$
	$\frac{1}{X_{cal}} \frac{G}{2\pi} \int_0^\infty \int_{-1}^1 j(E, \alpha) \int_0^{2\pi} \int_{t_1}^{t_2} H(\cos \theta(\alpha, \beta, t)) dt d\beta d \cos \alpha \varepsilon(E) dE$	$h_j^{(A)} = \frac{G\Delta\cos\alpha_j}{2\pi} \int_0^{2\pi} \int_{t_1}^{t_2} H(\cos \theta(\alpha_j, \beta, t)) dt d\beta \quad (\text{see note 1})$
	$\frac{1}{X_{cal}} \frac{G}{2\pi} \int_0^\infty \int_0^1 j(E, \alpha) \int_0^{2\pi} \int_{t_1}^{t_2} \left[ H(\cos \theta(\alpha, \beta, t)) + H(\cos \theta(\pi - \alpha, \beta, t)) \right] dt d\beta d \cos \alpha \varepsilon(E) dE$	$h_j^{(A)} = G(t_2 - t_1) \Delta \cos \alpha_j$
	$\frac{1}{X_{cal}} \frac{G}{2\pi} \int_0^\infty j(E) \int_{-1}^1 \int_0^{2\pi} \int_{t_1}^{t_2} H(\cos \theta(\alpha, \beta, t)) dt d\beta d \cos \alpha \varepsilon(E) dE$	$h^{(A)} = G(t_2 - t_1)$
Pinhole	$\frac{G}{X_{cal}} \int_0^\infty \left[ \int_{t_1}^{t_2} j(E, \alpha_0(t), \beta_0(t)) dt \right] \varepsilon(E) dE$	$h_{jk}^{(A)} = G \int_{t_1}^{t_2} \left\{ \begin{array}{lll} \frac{\alpha_0(t) - \alpha_{j-1}}{\alpha_j - \alpha_{j-1}} \frac{\beta_0(t) - \beta_{k-1}}{\beta_k - \beta_{k-1}} & \alpha_{j-1} < \alpha_0(t) < \alpha_j & \\ \frac{\alpha_0(t) - \alpha_{j-1}}{\alpha_{j+1} - \alpha_0(t)} \frac{\beta_{k+1} - \beta_0(t)}{\beta_k - \beta_{k-1}} & \beta_{k-1} < \beta_0(t) < \beta_k & \\ \frac{\alpha_j - \alpha_{j-1}}{\alpha_{j+1} - \alpha_0(t)} \frac{\beta_{k+1} - \beta_k}{\beta_0(t) - \beta_{k-1}} & \alpha_{j-1} < \alpha_0(t) < \alpha_j & \\ \frac{\alpha_j - \alpha_{j-1}}{\alpha_{j+1} - \alpha_0(t)} \frac{\beta_{k+1} - \beta_k}{\beta_0(t) - \beta_{k-1}} & \beta_k \leq \beta_0(t) < \beta_{k+1} & \\ \frac{\alpha_{j+1} - \alpha_0(t)}{\alpha_{j+1} - \alpha_j} \frac{\beta_k - \beta_{k-1}}{\beta_{k+1} - \beta_k} & \alpha_j \leq \alpha_0(t) < \alpha_{j+1} & \\ \frac{\alpha_{j+1} - \alpha_0(t)}{\alpha_{j+1} - \alpha_j} \frac{\beta_{k+1} - \beta_0(t)}{\beta_k - \beta_{k-1}} & \beta_{k-1} < \beta_0(t) < \beta_k & \\ \frac{\alpha_{j+1} - \alpha_j}{0} \frac{\beta_{k+1} - \beta_k}{\text{otherwise}} & \alpha_j \leq \alpha_0(t) < \alpha_{j+1} & \\ & \beta_k \leq \beta_0(t) < \beta_{k+1} & \end{array} \right\} dt$
	$\frac{G}{X_{cal}} \int_0^\infty \left[ \int_{t_1}^{t_2} j(E, \alpha_0(t)) dt \right] \varepsilon(E) dE$	$h_j^{(A)} = G \int_{t_1}^{t_2} \left\{ \begin{array}{ll} \frac{\alpha_0(t) - \alpha_{j-1}}{\alpha_j - \alpha_{j-1}} & \alpha_{j-1} < \alpha_0(t) < \alpha_j \\ \frac{\alpha_{j+1} - \alpha_0(t)}{\alpha_{j+1} - \alpha_j} & \alpha_j \leq \alpha_0(t) < \alpha_{j+1} \\ 0 & \text{otherwise} \end{array} \right\} dt$

Sensor Type	Integral for $\lambda$	Weights
	$\frac{G}{X_{cal}} \int_0^{\infty} j(E) \varepsilon(E) dE$	$h^{(A)} = G(t_2 - t_1)$
<b>Energy Response (see notes)</b>		
Arbitrary Energy Response	$\frac{1}{X_{cal}} \int_0^{\infty} \int_{-1}^1 \int_0^{2\pi} j(E, \alpha, \beta) \int_{t_1}^{t_2} A(\theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt d\beta d \cos \alpha \varepsilon(E) dE$	$h_i^{(E)} = \frac{\varepsilon(E_i) \Delta E_i}{X_{cal}}$
	$\frac{1}{X_{cal}} \int_0^{\infty} \varepsilon(E) dE \int_{-1}^1 \int_0^{2\pi} j(\alpha, \beta) \int_{t_1}^{t_2} A(\theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt d\beta d \cos \alpha$	$h_i^{(E)} = \frac{1}{X_{cal}} \int_0^{\infty} \varepsilon(E) dE$
(Narrow) differential channel	$\frac{\varepsilon \Delta E}{X_{cal}} \int_{-1}^1 \int_0^{2\pi} j(E_0, \alpha, \beta) \int_{t_1}^{t_2} A(\theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt d\beta d \cos \alpha$	$h_i^{(E)} = \frac{\varepsilon \Delta E}{X_{cal}} \begin{cases} \frac{E_0 - E_{i-1}}{E_i - E_{i-1}} & E_{i-1} < E_0 < E_i \\ \frac{E_{i+1} - E_0}{E_{i+1} - E_i} & E_i \leq E_0 < E_{i+1} \\ 0 & \text{otherwise} \end{cases}$
	$\frac{\varepsilon \Delta E}{X_{cal}} \int_{-1}^1 \int_0^{2\pi} j(\alpha, \beta) \int_{t_1}^{t_2} A(\theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt d\beta d \cos \alpha$	$h^{(E)} = \frac{\varepsilon \Delta E}{X_{cal}}$

Sensor Type	Integral for $\lambda$	Weights
Wide (differential) channel	$\frac{\varepsilon}{X_{cal}} \int_{E_0-1}^{E_1} \int_{-1}^1 \int_0^{2\pi} j(E, \alpha, \beta) \int_{t_1}^{t_2} A(\theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt d\beta d \cos \alpha dE$	$h_i^{(E)} = \frac{\varepsilon}{X_{cal}} \left[ \left\{ \begin{array}{ll} 0 & E_0 > E_{i+1} \\ \Delta E_i - \frac{(E_0 - E_{i-1})^2}{2(E_i - E_{i-1})} & E_{i-1} < E_0 < E_i \\ \frac{(E_{i+1} - E_0)^2}{2(E_{i+1} - E_i)} & E_i \leq E_0 < E_{i+1} \\ \Delta E_i & E_0 < E_{i-1} \end{array} \right\} - \left\{ \begin{array}{ll} 0 & E_1 > E_{i+1} \\ \Delta E_i - \frac{(E_1 - E_{i-1})^2}{2(E_i - E_{i-1})} & E_{i-1} < E_1 < E_i \\ \frac{(E_{i+1} - E_1)^2}{2(E_{i+1} - E_i)} & E_i \leq E_1 < E_{i+1} \\ \Delta E_i & E_1 < E_{i-1} \end{array} \right\} \right]$
	$\frac{\varepsilon}{X_{cal}} \int_{E_0}^{E_1} dE \int_{-1}^1 \int_0^{2\pi} j(\alpha, \beta) \int_{t_1}^{t_2} A(\theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt d\beta d \cos \alpha$	$h^{(E)} = \frac{\varepsilon(E_2 - E_1)}{X_{cal}}$
Integral channel	$\frac{\varepsilon}{X_{cal}} \int_{E_0-1}^{\infty} \int_{-1}^1 \int_0^{2\pi} j(E, \alpha, \beta) \int_{t_1}^{t_2} A(\theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt d\beta d \cos \alpha dE$	$h_i^{(E)} = \frac{\varepsilon}{X_{cal}} \left\{ \begin{array}{ll} 0 & E_0 > E_{i+1} \\ \Delta E_i - \frac{(E_0 - E_{i-1})^2}{2(E_i - E_{i-1})} & E_{i-1} < E_0 < E_i \\ \frac{(E_{i+1} - E_0)^2}{2(E_{i+1} - E_i)} & E_i \leq E_0 < E_{i+1} \\ \Delta E_i & E_0 < E_{i-1} \end{array} \right.$
	$\frac{\varepsilon}{X_{cal}} \int_{E_0}^{\infty} dE \int_{-1}^1 \int_0^{2\pi} j(\alpha, \beta) \int_{t_1}^{t_2} A(\theta(\alpha, \beta, t), \phi(\alpha, \beta, t)) dt d\beta d \cos \alpha$	$h^{(E)} = \infty \text{ (library supplies } \varepsilon/X_{cal} \text{ instead of infinity)}$

Sensor Type	Integral for $\lambda$	Weights
<p>*Notes:</p> <ol style="list-style-type: none"> <li>1. For cylindrically symmetric sensors, with the gyrotropic assumption, an integral over <math>\beta</math> implies that <math>\beta_0</math> and <math>\phi_b</math> need not be known.</li> <li>2. For separable energy/angle response, we define <math>h_{ijk} = h_i^{(E)} h_{jk}^{(A)}</math>, or <math>h_{ij} = h_i^{(E)} h_j^{(A)}</math> in the gyrotropic case. <math>X_{cal}</math> goes into the Energy weights, and time goes into the angular weights.</li> <li>3. <math>H(x)</math> is the Heaviside step function, with value 0 for <math>x &lt; 0</math> and 1 for <math>x \geq 0</math>.</li> </ol>		

The formula for the trapezoidal integral weights on a grid from  $x_I$  to  $x_N$  is:

$$\Delta x_i = \begin{cases} \frac{x_2 - x_1}{2} & i = 1 \\ \frac{x_N - x_{N-1}}{2} & i = N \\ \frac{x_{i+1} - x_{i-1}}{2} & else \end{cases}$$