Appendix B: Converting from (Φ, *K*) to (*L*, *B*) Coordinates and Vice-versa

B.1 Introduction

Because the *K*-Φ mapping has not been widely used in the past, it is very useful to be able to transform to and from *K*-Φ and the more traditional *B*-*L* system. Here *B*-*L* is used generically to refer to different transforms of *B* (e.g., *B*, B/B_0, etc.) and both L_m and L^*.

Unfortunately, to date there is no simple transformation between K-h_{min} and *B*-*L*.

This appendix discusses the conversion between (Φ, *K*) and (*L*, *B*) coordinates. The following definitions are noted:

- *B* refers to B_m, the magnetic field at the mirror point.
- B_{eq} is the magnetic field at the magnetic equator, also referred to as B_{min}, the minimum magnetic field on the field line.
- B_0 is the approximate value of B_{eq} based on the McIlwain definition – see equation (B5).
- The term (*L*, *B*) means a generic coordinate system which refers specifically to the McIlwain L_m,. Here, *B* can refer to B_m, B_m/B_{eq}, or B_m/B_0.

We generally know either (Φ, *K*) or (*L*, *B*). We want to be able to convert from one to the other, and we especially need to know the equatorial pitch angle α_{eq}:

$$\alpha_{eq} = \sin^{-1}\sqrt{\frac{B_{\min}}{B_m}}$$

or (B1)

$$y = \sin(\alpha_{eq}) = \sqrt{\frac{B_{\min}}{B_m}}$$

B.2 Method 1 (Hilton)

In order to convert from (Φ, *K*) to (*L*, *B*) coordinates or from (*L*, *B*) to (Φ, *K*), we first note some basic definitions.

Roederer's L^*:

$$L^* = \frac{2\pi k_0}{\Phi} = \frac{\mu_0 M_E}{2\Phi} \tag{B2}$$

where M_E is the Earth magnetic dipole moment, μ_0 is the permeability of free space, and k_0 is the magnetic dipole parameter:

$$k_0 = \frac{\mu_0 M_E}{4\pi} \tag{B3}$$

Kaufmann's *K* [*Kaufmann*, 1965]:

$$\begin{aligned} K &\equiv \frac{J}{\sqrt{8 m_0 \mu}} = I\sqrt{B_m} \\ &= \int_A^{A'} \left[B_m - B(s)\right]^{1/2} ds \end{aligned} \tag{B4}$$

We also have to make a couple of approximations:

$$B_{eq} \approx B_0 = \frac{k_0}{L_m^3};\quad k_0 \equiv 0.311653 \text{ G-R}_E^3 \tag{B5}$$

$$L_m \approx L^* \tag{B6}$$

Finally, we use the approximation of *Hilton* [1971] to obtain L_m as a function of *B* and *I*:

$$X = I^3 B_m / k_0 \tag{B7}$$

$$v = X^{1/3} \tag{B8}$$

$$F = L_m^3 B_m / k_0 = a_1 v^3 + a_2 v^2 + a_3 v + a_4 \tag{B9}$$

With these relations in hand, the procedure to go from (Φ, *K*) to (L_m, B/B_0) is as follows:

1) Compute $L^* \approx L_m$ using (B2).
2) Knowing L_m and *K*, compute B_m using an iterative procedure:
    a) Assume a value of B_m.
    b) From B_m and *K*, compute *I* from (B4).
    c) From *I* and B_m use Hilton's function to obtain L_m.
    d) Iterate on B_m until L_m converges on the input value.
3) Compute B_0 from (B5). We then know B_m/B_0.

To go from(L_m, B/B_0) to (Φ, *K*) we reverse the process:

1) Knowing L_m compute B_0 from (B5), then compute B.

2) Knowing L_m and B, invert Hilton's function to obtain I. (NOTE: since Hilton's function is a polynomial in v, it's relatively easy to find the roots.)

3) Knowing I and B, compute K.

4) Again assuming $L^* \approx L_m$, compute Φ using (B1).

B.3 Method 2 (Schulz and Lanzerotti)

For a dipole field, we can get the following from *Schulz and Lanzerotti* [1974]:

$$K = \frac{Y(y)}{y}\sqrt{\frac{\Phi}{2\pi}} \tag{B10}$$

where $Y(y)$ is a function for which several approximations are available. Generally, this doesn't do us much good by itself because we generally know Φ and K and want to find y. Paul O'Brien has provided a way to invert (B10) to get y as a function of Φ and K.

B.4 Summary

If we know L_m, B_m, B_{eq}, Φ, and K, we're home free: we can derive virtually any other needed property of the particles. Otherwise we need to make some assumptions or approximations.

- Method 1 relies on the approximations $B_{eq} \approx B_0$ and $L_m \approx L^*$. Some quick calculations indicate that the former is good to within about 5% out to $B_m / B_{eq} \approx 2$; B_0 is generally smaller than B_{eq}. The latter seems to be good to within about 5%.

- Method 2 relies on the dipole approximation

B.5 Implementation

A package of MATLAB routines has been written to implement these procedures.

`function Lm=hiltonBI(B, I, k0)` implements Hilton's function to find L_m given B and I – equations (6) – (8).

`function [I B] = findIB(K, L, k0)` implements the procedure to go from (Φ, K) to (L_m, B/B_0).

`function I=hiltonBL(B, Lm, k0)` implements the inverse of Hilton's function to find I given L_m and B. All that remains is to compute K from I and B and Φ from $L^* \approx L_m$.

`function y = K_to_y(K, Phi)` implements Paul's procedure to find the inverse of (9).