

# 1.5 Fast Coordinates Used for Invariant Coordinate Calculations

---

This section describes the neural networks used to quickly compute charged particle drift invariant coordinates for use in AE9/AP9/SPM V1.0/V1.1. Specifically, neural networks were trained to rapidly compute the  $\Phi$  and  $h_{min}$  drift invariants in the Olson-Pfizer Quiet magnetic field model [Olson and Pfizer, 1977]. These neural networks contributed to a dramatic and necessary increase in the speed of the AE9/AP9/SPM calculations, since it is necessary to perform several coordinate conversions at runtime for every ephemeris point requested by the user.

This section has also been released as Aerospace Report No. TOR-2014-00293. Matlab is a trademark of The Mathworks Inc.

## 1.5.1 Introduction

The magnetic coordinates used in the AE9/AP9/SPM model (see Section 1.2) can be time consuming to compute. The field-line coordinates  $L_m$  and  $\alpha_{eq}$  require a trace of the local field line, while the drift invariants  $\Phi$  and  $h_{min}$  require a trace of the entire drift shell. The drift invariants, in particular, take far too long to compute for a practical runtime calculation. We have, therefore, utilized two techniques to speed up the calculations. First, we employ a fast field line trace [Pfizer, 1991, 1995] that (1) computes the invariant integral  $I$  for multiple initial local pitch angles, and (2) truncates the order of the internal field model (IGRF) [Finlay et al., 2010] as a function of altitude. For the Olson-Pfizer Quiet model [Olson and Pfizer, 1977], these speed-ups in the local field line trace improve upon the IRBEM library [IRBEM, 2012] by a factor of 20-30. Second, we utilize neural networks to capture pre-computed  $\Phi$  and  $h_{min}$  as a function of time,  $I$ , and  $B_m$  (the mirror field strength). This second speed-up still requires a trace of the local field line, but replaces the drift shell trace with an evaluation of the neural network. The use of neural networks combined with the fast field line trace nets a total speed up factor of 250-1300 relative to the full drift orbit trace implemented in the IRBEM library. Details of these two speed-up techniques follow.

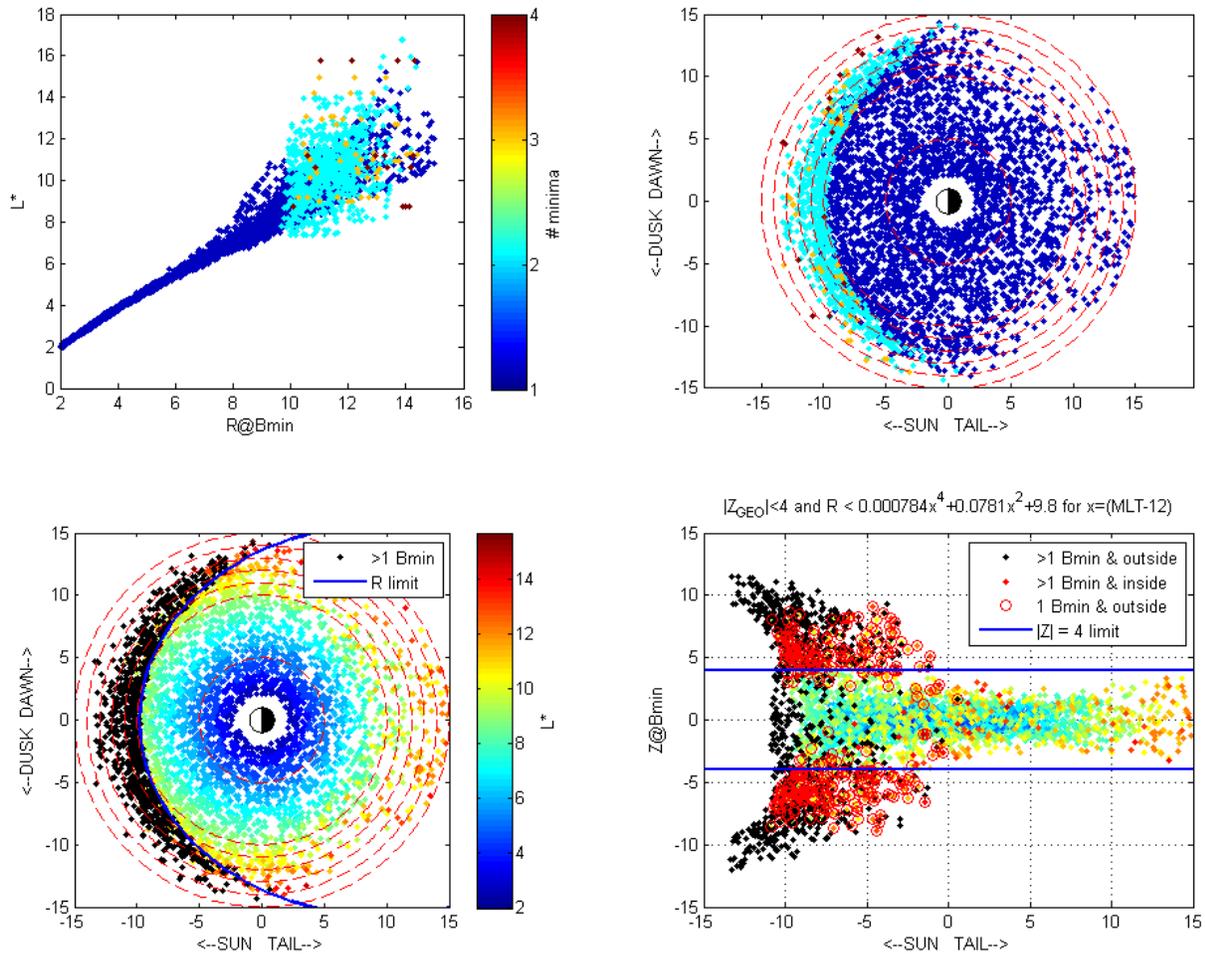
We note that the use of neural networks to avoid the full drift invariant calculation was developed simultaneously by our partners at Los Alamos National Laboratory for a dynamic external field model [Koller et al., 2009; Koller and Zaharia, 2011]. Our networks are similar in concept. However, they address a simpler, quiet magnetic field model but have had to meet tighter requirements in terms of error performance and boundary definitions.

## 1.5.2 Fast Field Line Trace

To accelerate the calculation of local field line coordinates, a fast field-line tracer is used to calculate the  $I$  integral. This is a modified version of the INVARM routines by Pfizer [1991; 1995], and provides typically a factor of 20-30 speed improvement over the counterpart routines in the IRBEM library. INVARM calculates  $I$  for multiple initial pitch angles simultaneously and

it truncates the IGRF expansion at a low order that is determined based on the geocentric distance (equivalent to altitude). The truncation is known within INVARM as SPIGRF, for “speed IGRF.” Limits on INVARM are: magnetic latitude  $< 75^\circ$ ; geocentric radius  $< 12 R_E$ ; and  $B < 0.00025$  G.

We modified the tracing routines to impose limits on the equatorial (minimum  $B$ ) location to exclude regions of Shabansky (bifurcated) drift orbits from the models. Such orbits are unstable, and therefore cannot be represented in the AE9/AP9/SPM invariant coordinate system. Figure 6 shows several diagnostics that were used to develop the Shabansky limits. The Shabansky limit is twofold: the geocentric radius of the equatorial location must be less than a polynomial in hours of local time past noon (pre-noon times are negative), and the  $Z$  component of the equatorial location in geographic coordinates must not exceed a constant (namely,  $4 R_E$ ). Points outside either of these limits are treated as being outside the model. This Shabansky classification is incorrect only a few percent of the time (depending on what kind of spatial



**Figure 6. Shabansky orbit diagnostics. Top left: number of magnetic field minima as a function of  $L^*$  and the radial distance at the minimum  $B$  ( $B_{min}$ ) point. Top right: the same versus local time coordinates of  $B_{min}$ . Bottom left:  $L^*$  versus local time of min, with a radial distance filter applied on the number of  $B_{min}$ . Bottom right: side view with various markings for points inside and outside the Shabansky limits.**

sampling is done). A de facto outer limit exists where the field model exceeds internal limits built into the INVARM tracing routines by its authors. The original field model limits and our new Shabansky limits are coded directly into the magnetic field tracing routines.

### 1.5.3 Fast $\Phi$ Neural Network

For the Olson-Fitzner Quiet external magnetic field [Olson and Pfizter, 1977] model added to the IGRF internal field model, the global magnetic field state is fully specified by the date – which incorporates information about the secular variation of the internal field and the seasonal and diurnal variation of the external field. For a fixed field state, the drift orbit is fully determined by McIlwain’s integral invariant  $I$  and the mirror magnetic field strength  $B_m$ . A neural network is trained to reproduce either  $\Phi$  or  $h_{min}$  (see below) given the date (as decimal modified Julian date, or MJD), as well as  $I$  and  $B_m$ .

Both neural networks require five inputs: (1)  $I^{1/4}$ , for  $I$  in  $R_E$ , (2)  $\log_{10}(B_m)$ , for  $B_m$  in nT, (3) MJD, (4) *YearPhase*, which is given below, and (5) UT as a fraction of a day. *YearPhase* is roughly a sidereal day of year, using 1/1/1950 as the reference phase zero and approximating a sidereal year as 365.25 days:

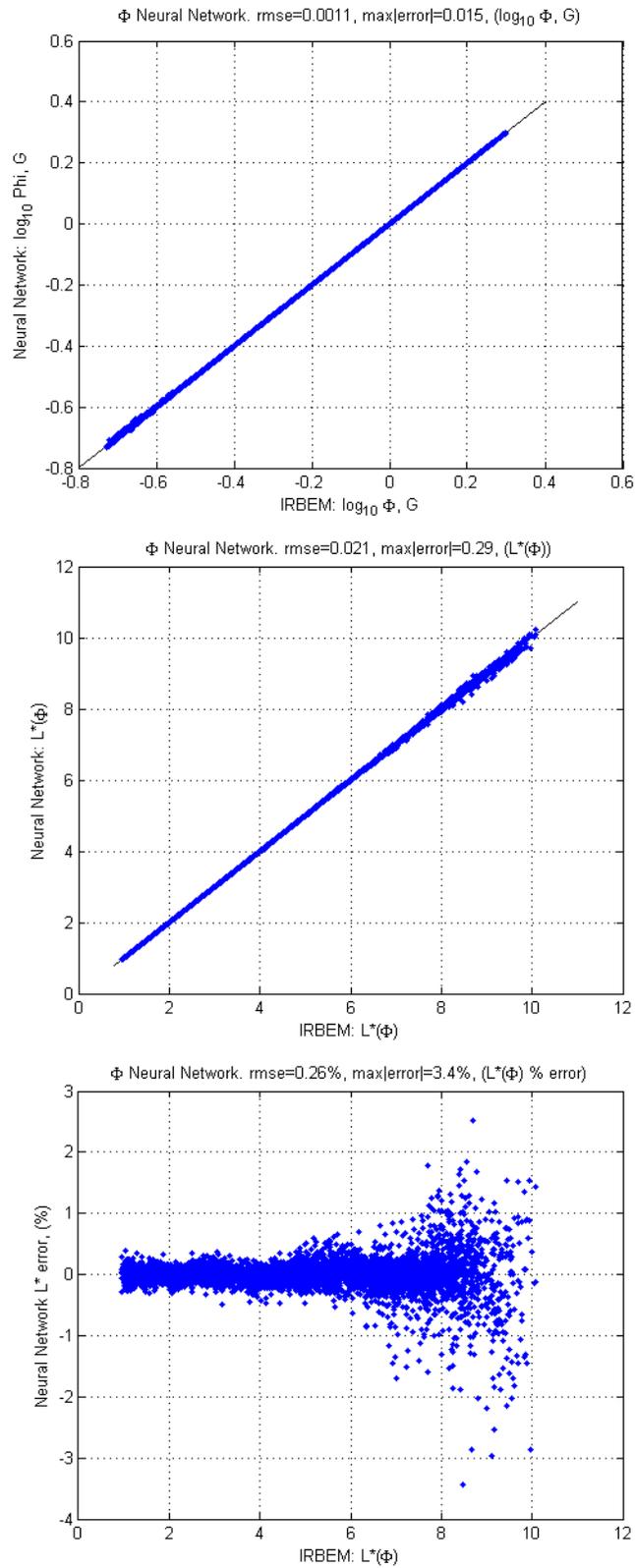
$$YearPhase = \text{rem}[\text{MJD} - \text{MJD}(1950,1,1), 365.25], \quad (76)$$

where  $\text{rem}(x,y)$  is the remainder of  $x$  divided by  $y$ . The  $\Phi$  network returns  $\log_{10}(\Phi)$  for  $\Phi$  in  $\text{GR}_E^2$ . The neural networks were developed with and conform to the open-source IRBEM neural network library (<http://irbem.svn.sourceforge.net/viewvc/irbem/extras/nlib/doc/nlib.pdf>).

The  $\Phi$  neural network was trained on about 64,000 points, where  $\Phi$  is computed from the IRBEM library. The points were sampled randomly from the 3-D space near the Earth, where it was assumed that each point was the mirror point of a particle. Because several of the inputs are periodic, we included in the training set duplicate points just beyond the periodic boundaries (e.g., UT just below zero and just above 1). The neural network library allows us to specify the error for each point. We chose the larger of 0.02 or  $0.01L^*$  as the error in  $L^*$  and then transformed that error into its equivalent for  $\log_{10}\Phi$ . The trained result of the neural network is stored in a stand-alone Matlab save set `fastPhi_net.mat`.

To validate the neural network, we generated an independent set of about 7,000 test points. We evaluate the neural network at each of these points, and computed some error statistics. Figure 7 shows three different approaches to the error statistics of the neural network. The most intuitive of these is the percent error in  $L^*$ , which is 0.26%, root mean squared error, and the largest  $L^*$  error is 3.5% beyond  $L^*=8$ . Thus, the fast  $\Phi$  neural network introduces negligible errors relative to the full drift orbit trace. We note that the actual error in the physical quantities underlying the  $L^*$  or  $\Phi$  calculation have errors that are much larger because the quiet magnetic field model is imperfect. However, the neural network’s purpose as a replacement for the full drift integral requires that it be a nearly identical mathematical replacement, regardless of the shortcomings of the magnetic field model from which it is derived.

Because the neural network is simply an analytical fit, it has no inherent ability to honor the limits of the input or output domain. Thus, without imposing limits, the neural network will



**Figure 7. Performance of neural network for computing  $\Phi$ . Top: Neural Network  $\log_{10}\Phi$  versus same for IRBEM library. Middle: Neural Network  $L^*$  versus same for IRBEM. Bottom: percent error in Neural Network  $L^*$ .**

return invalid  $\Phi$  when given input parameters outside the training domain – that is, input parameters corresponding to the atmospheric loss cone or regions outside the radiation belts, or times beyond the validity of the IGRF data. To solve this problem, we impose a set of boundary conditions on the models: (1) we limit MJD to 1 Jan 2015, (2) we treat as “outside the model” points for which  $\log_{10}B_m$  exceeds a prescribed polynomial in  $I^{1/4}$ , or (3) when  $I^{1/4}$  exceeds a maximum value of  $2.23 R_E^{1/4}$ . For the  $\Phi$  network, the  $\log_{10}B_m$  is shown in Figure 8. The loss cone limit is determine on the surface  $h_{min} = 100$  km because any point reaching that low will actually end up being covered by the  $h_{min}$  grid in the AE9/AP9/SPM models, and therefore will not need  $\Phi$  at all.

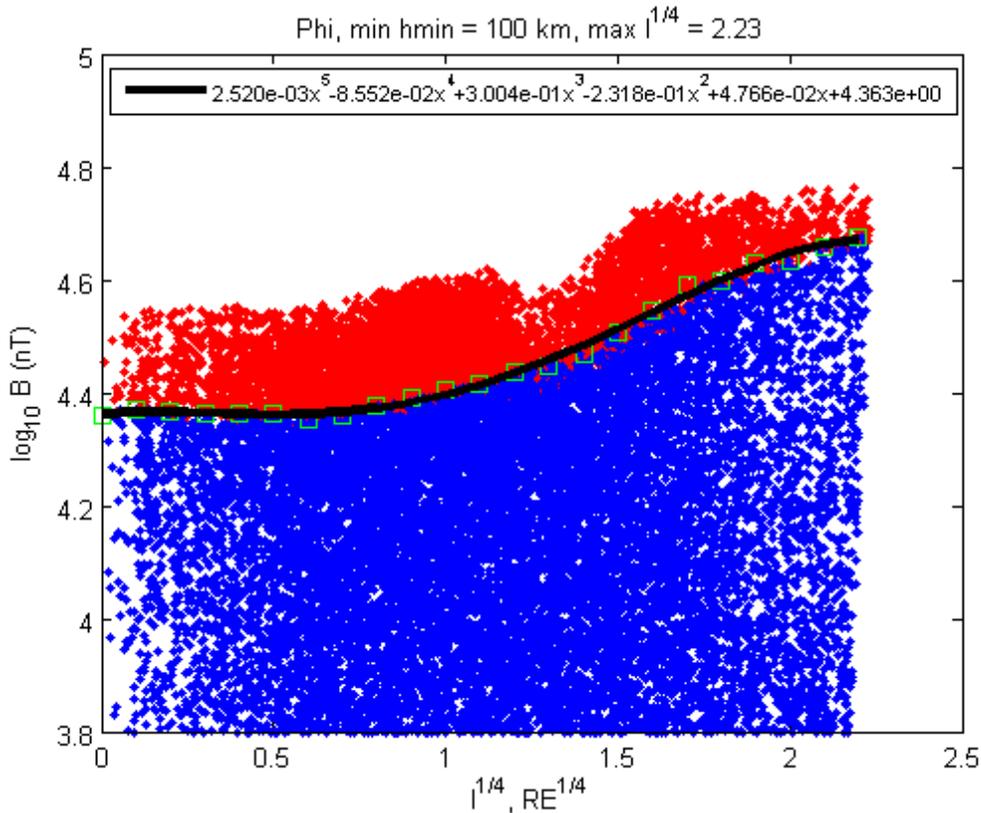


Figure 8. Depiction of the boundaries of the  $\Phi$  neural network. Red points indicate those for which the minimum altitude reached by the particle (its  $h_{min}$ ) is below 100 km, and is thus in the loss cone of the  $\Phi$  grid of the AE9/AP9 models. The green boxes indicate values of maximum  $\log_{10}B$  (the mirror field strength, or  $B_m$ ) in bins of  $I^{1/4}$ . A polynomial in  $I^{1/4}$  is used to define the loss cone, while the maximum  $I^{1/4}$  is used to define the trapping limit.

### 1.5.4 Fast $h_{min}$ Neural Network

The  $h_{min}$  network is structurally identical to the  $\Phi$  neural network, except it returns  $h_{min}$  in km. Although  $h_{min}$  is used only as a coordinate for particles mirroring at low altitudes, in order to determine whether to use the  $K$ - $\Phi$  or  $K$ - $h_{min}$  grid, we must first evaluate  $h_{min}$  for any point,

regardless of altitude. Therefore, the  $h_{min}$  neural networks covers the entire 3-D domain covered by the  $\Phi$  network, as well as a lower altitude portion that actually extends into the Earth, slightly.

The internal components of the IGRF introduce considerable structure in  $h_{min}$  for particles that reach low altitudes, or whose orbits will eventually take them below the surface of the Earth. Therefore, a larger training set was needed, about 168,000 points. Similarly, a larger validation set was used, about 18,000 points. All points were treated as having the same error.

The error performance of the  $h_{min}$  network is depicted in Figure 9. The root mean squared error is 25 km, and the maximum error is 329 km, and such large errors occur only at large positive or negative  $h_{min}$ , i.e., outside the domain of the AE9/AP9  $K-h_{min}$  grids. Looking only at the range from -600 km to +1200 km (i.e., just encompassing the  $K-h_{min}$  range of AE9/AP9's low altitude grid), the rms error is 12 km, and the maximum error is 80 km. An investigation of these errors revealed that they were actually the result of limitations of the IRBEM lib itself – it was never designed to trace accurately inside the Earth.

Like the  $\Phi$  neural network, the  $h_{min}$  network requires definitions of the valid limits of inputs. The same scheme is used, specifying limits on MJD,  $B_m$  and  $I$ . The limits are shown in Figure 10. It is noteworthy that the loss cone for the  $h_{min}$  network is defined as *negative* 500 km. This is because the AE9 model includes points in the “drift loss cone”; that is, they are trapped on the local field line, but will encounter the atmosphere before the complete a full drift orbit.

### 1.5.5 Known Issues

The primary “known issue” for the fast neural networks and for the field model in general is extrapolation of the IGRF beyond the year 2015. Some kind of extrapolation is certainly possible, but it is not yet known which moments of the IGRF can be extrapolated, and how far out into the future.

A secondary “known issue” is the limitation on the  $h_{min}$  network at  $h_{min} < -500$  km. This is actually an issue as much with the IRBEM library as it is with the neural network. It can likely be resolved in the future using a fixed-precision tracing algorithm rather than IRBEM's fixed-time algorithm.

### 1.5.6 Summary

We have presented details of two different components of our efforts to improve the run time of AE9/AP9/SPM by speeding up the calculation of magnetic invariant coordinates. These speed-ups include a fast field line tracer that was developed by *Pfizer* [1991, 1995], and neural networks to replace drift orbit tracing. Together, these speed-ups provide a factor of ~1000 performance improvement over the IRBEM library tracing routines.

We have shown that the neural networks perform very well out of sample, and are adequate replacements for the full numerical integrals. We have also shown that the switch from a numerical integration to the neural networks required some additional constraints to identify the loss cone, Shabansky orbits, and the domain of validity.

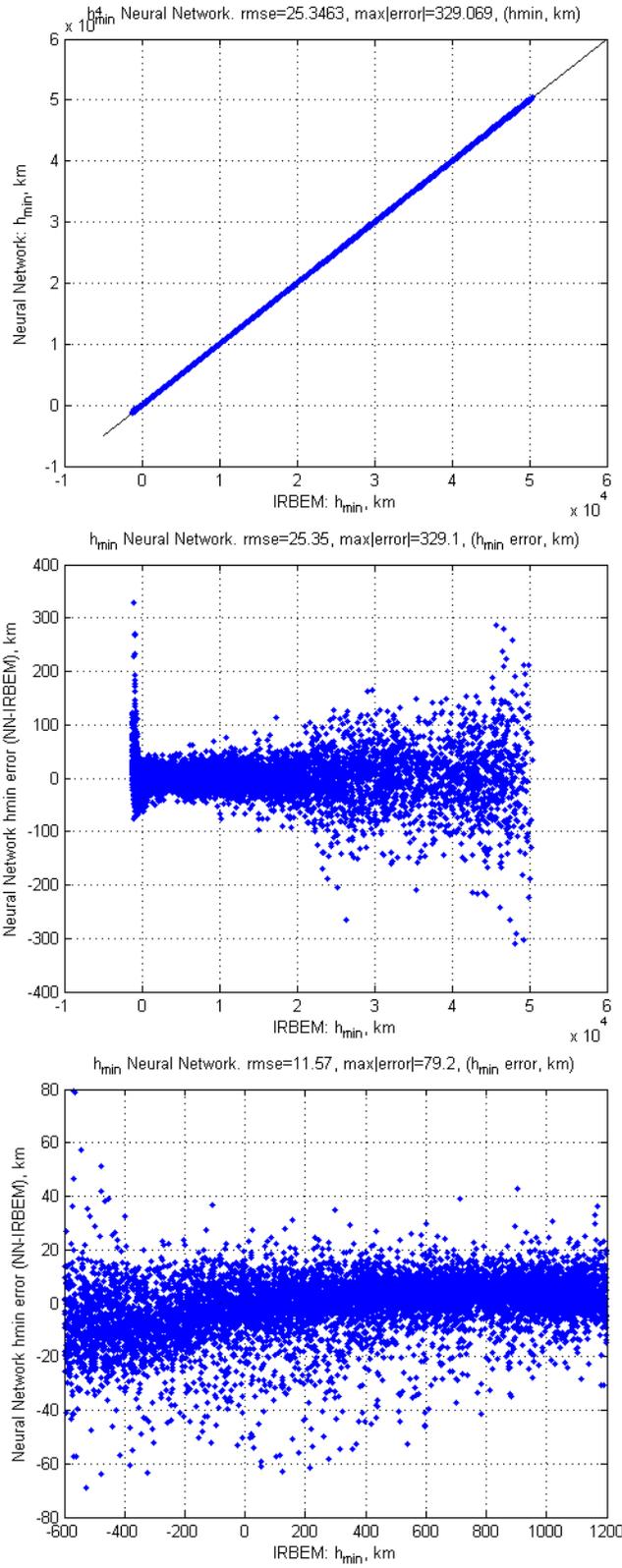


Figure 9. Error performance of the  $h_{min}$  neural network (NN). Top: NN  $h_{min}$  versus the IRBEM library. Middle: Neural network error over the full domain. Bottom: NN error over the low altitude grid region.

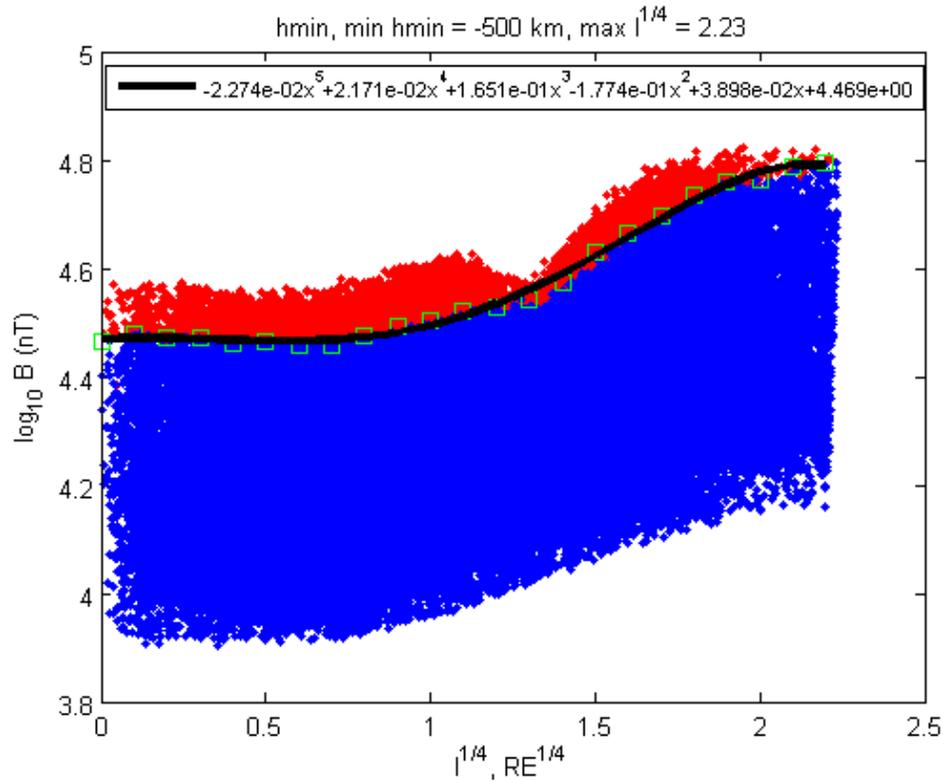


Figure 10. The boundaries of the  $h_{min}$  neural network. Red points indicate those for which the minimum altitude reached by the particle (its  $h_{min}$ ) is below -500 km, and is thus in the drift loss cone. The green boxes indicate values of maximum  $\log_{10}B$  (the mirror field strength, or  $B_m$ ) in bins of  $l^{1/4}$ . A polynomial in  $l^{1/4}$  is used to define the loss cone, while the maximum  $l^{1/4}$  is used to define the trapping limit.