Oulu - Open-source Geomagnetosphere Propagation Tool Python Package (OTSOpy) User Manual

Nicholas Larsen

May 5, 2025

Abstract

This manual outlines, in detail, all of the necessary information needed by a user to use the "Oulu — Open-source Geomagnetosphere Propagation Tool" Python Package (OTSOpy). It explains the various functions that OTSOpy provides, including the input parameters and options available to the user. For more in depth detail on the OTSO tool and its internal working please see Larsen et al. (2023).

1 Introduction

OTSO is a tool that performs the numerical integration of the equations of motion for a charged particle within the Earth's magnetic field. This is done to simulate the trajectory of cosmic rays when they encounter the magnetosphere and to determine where said cosmic rays encounter the Earth's surface. OTSO can do this for single cosmic rays to determine the trajectory or for many cosmic rays of varying rigidities to determine important parameters, such as cut-off rigidities and asymptotic cones. As such, this tool is primarily designed to aid in cosmic ray research and the study of space weather events, namely ground-level enhancements OTSO is written in two different programming languages, Python and FORTRAN. Python is used to enter the input parameters and run the program. The Fortran section is responsible for the numerical integration and modelling of the Earth's magnetic field. This combination mixes the user-friendly nature of Python with the older, more efficient, which has access to many previously created open-access libraries designed for this field.

OTSO is designed to be open-source. The user of OTSO has free rein to edit the program as they see fit to achieve their goals. Additions to the base tool can be made by the community, which will help develop the tool into a robust geomagnetic computation tool to aid the cosmic ray research community. OTSOpy has pre-compiled Fortran libraries included for ease of distribution and use. If the user wishes to make edits to the Fortran code themselves, they must recompile the Fortran library. To do this, users should follow the steps outlined in the original OTSO (see `https://github.com/NLarsen15/OTSO`) release, but apply the workflow to the Fortran code included in the OTSOpy repository and not the older OTSO repository.

As new functions are added, I will endeavour to update this manual and the supplied scripts to include information on the new applications as frequently as possible. I, however, implore those who made additional edits that are added to the base tool to provide their own documentation detailing the new features to the repository to simplify this process. OTSOpy is more developed than the original OTSO tool, and though I will try to keep the original OTSO up to date, the original OTSO repository will likely be superseded by the OTSOpy repository as it is easier to develop for, and much easier to use for most people, as no compilation or Fortran is needed to run the package.

2 Installation

Installation is simple and can be approached in two ways: either using pip or cloning the repository. Detailed instructions and commands are given within the repository home page. See `https://github.com/NLarsen15/OTSOpy` for details.

3 Input Parameters

This section lists all the configurable input parameters used across OTSOpy functions. Each parameter includes its type, default value, and description. Refer to this section when setting up parameters for specific function calls.

3.1 General Settings

- **Stations**
  `Type:` list of strings
  `Default: None`
  List of neutron monitor station names to include in the simulation. A CSV file containing common station names and acronyms is included in the package. If the station name is not found you may add the station to the CSV or use the customlocations option.

- **customlocations**
  `Type:` list of lists [[ ]]
  `Default: None`
  Custom location(s) in the format `[name, latitude, longitude]`
  `name Type:` string
  `latitude Type:` float
  `longitude Type:` float
  Allows users to enter custom locations around the Earth and name the location as they wish. Useful for testing prospective locations for neutron monitors or for neutron monitors that have been overlooked and aren't included in OTSO currently.

- **Locations**
  `Type:` list of lists [[ ]]
  `Default: None`
  Custom location(s) in the format `[X, Y, Z]`. `X,Y,Z Type:` float
  For use in the coordinate transform and magnetic field functions, this is a list of lists where the nested list contains positional information of the location that you wish to convert to a different coordinate system or know the magnetic field vector at that location.

- **latitudes, longitudes, altitudes**
  `Type:` list of floats
  `Default: None`
  Arguments used in the flight function to define the positional information of the path along which computations should be done.

- **rigidity**
  `Type:` float
  `Default: None`

Rigidity of the CR in gigavolts [GV].

- **rigiditystep**
  `Type:` string
  `Default: "ON"`
  `Options: "ON" / "OFF"`
  Can be invoked so OTSO will perform a low-resolution scan of the input range of rigidities to obtain rough estimates of the upper and lower cut-offs, which are used to define a smaller range of rigidities to iterate over. This option speeds up cut-off computations significantly.

- **startrigidity, endrigidity, rigiditystep**
  `Type:` float
  `Default: 20, 0, 0.01`
  Start and end rigidity define the range of rigidities over which trajectory computations will be done. The `rigiditystep` variable sets the resolution of the computation. Values are in gigavolts [GV]. For example, under default settings, the trajectory of particles with rigidities 20,19.99,19.98,...,0 will be computed.

- **cutoff_comp**
  `Type:` string
  `Default: "Vertical"`
  `Options: "Vertical", "Apparent", "Custom"`
  Defines what cut-off computation you wish to perform, as the effective cut-off is azimuth and zenith dependent. `"Vertical"` computes the effective cut-off along the zenith; `"Apparent"` computes the apparent effective cut-off by finding a weighted mean over nine unique zenith and azimuth combinations as defined by Bieber et al. (1997); and `Custom` takes the user inputted zenith and azimuth and computes the effective cut-off rigidity along that starting direction.

3.2 Time and Date Settings

- **year, month, day, hour, minute, second**
  `Type:` integers
  `Default: 2024, 1, 1, 12, 0, 0`
  Specifies the date and time at which the simulation begins.

- **dates**
  `Type:` list of datetimes
  `Default: None`
  Used within the coordinate transformation and flight functions, `date` specifies the date and time for the computation.

3.3 Initial Conditions

- **startaltitude**
  `Type:` float
  `Default: 20`
  Starting altitude [km].

- **zenith, azimuth**
  `Type:` float
  `Default: 0, 0`
  Starting zenith and azimuth angles in degrees.

3.4 Space Weather / Solar Wind Settings

- **vx, vy, vz**
  `Type:` float
  `Default: -500, 0, 0`
  GSM Velocity vector components of the solar wind [km/s].

- **by, bz**
  `Type:` float
  `Default: 5, 5`
  Interplanetary magnetic field components [nT].

- **density**
  `Type:` float
  `Default: 1`
  Solar wind density cm^{-3}.

- **pdyn**
  `Type:` float
  `Default: 0`
  Solar wind dynamic pressure [nPa]. If set to `0`, then pdyn is computed using density and solar wind velocity values. If set as not `0`, then the inputted pydn will overwrite the derived pdyn from the solar wind density and velocity.

3.5 Magnetic Field Settings

- **internalmag**
  `Type:` string
  `Default: "IGRF"`
  `Options: "NONE", "IGRF", "Dipole", "Custom Gauss"`
  Internal magnetic field model to be used (e.g., `"IGRF"`). Users may select `"NONE"` for no internal magnetic field, `"IGRF"` for the IGRF model, `"Dipole"` for a simplified dipole model, and `"Custom Gauss"` for user-inputted Gaussian coefficients (see variables **g** and **h**).

- **externalmag**
  `Type:` string
  `Default: "TSY89"`
  `Options: "NONE", "TSY87short", "TSY87long", "TSY89", "TSY89_BOBERG", "TSY96", "TSY01", "TSY01S", "TSY04", "MHD"`
  External magnetic field model to be used. `"NONE"` is no external magnetic field, `"TSY87short"` and `"TSY87long"` are the long and short variants of the Tsyganenko 1987 model Tsyganenko (1987), `"TSY89"` is the Tsyganenko 1989 model Tsyganenko (1989), `"TSY89_BOBERG"` is the Boberg extension of the Tsyganenko 1989 model Boberg et al. (1995), `"TSY96"` is the Tsyganenko 1996 model Tsyganenko (1995), `"TSY01"` is the Tsyganenko 2001 model Tsyganenko (2002a,b), `"TSY01S"` is the extension of the Tsyganenko 2001 model to extreme geomagnetic disturbances Tsyganenko et al. (2003), and `"TSY04"` is the Tsyganenko 2004 model Tsyganenko and Sitnov (2005). `"MHD"` is an option for including a matrix grid of positional and magnetic field data (not necessarily generated by MHD simulations) that can be read into OTSO and used.

- **g, h**
  `Type:` list of floats
  `Default: None`
  Optional Gauss coefficients for the magnetic field. The length of each list should be 105, matching the number of coefficients for a harmonic expansion up to the 13^{th} degree.

- **MHDfile, MHDcoordsys**
  `Type:` string
  `Default: None`
  MHDfile is the path to the file containing the positional and magnetic data. NOTE that the file

needs to be a CSV file and the columns organised as such `X,Y,Z,Bx,By,Bz`. The grid points also need to be evenly spaced, or the computation will fail. The MHDcoordsys system specifies the coordinate system of the matrix data.

3.6 Geomagnetic Indices

- **kp**
  `Type:` integer
  `Default: 0`
  Planetary K index.

- **Dst**
  `Type:` float
  `Default: 0`
  Disturbance storm time index (Dst) index value.

- **G1–G3**
  `Type:` float
  `Default: 0`
  TSY01 and TSY01S specific parameters are determined by solar wind conditions during the hour preceding the desired computation time. For details see Tsyganenko (2002a,b) and Tsyganenko et al. (2003).

- **W1–W6**
  `Type:` float
  `Default: 0`
  TSY04 specific parameters that quantify the strength of six main magnetospheric currents. These can be found at `https://geo.phys.spbu.ru/~tsyganenko/models/ts05/` or they can be computed by the OTSO serverdata function using OMNI data.

3.7 Integration and Computation Termination Settings

- **intmodel**
  `Type:` string
  `Default: "Boris"`
  `Options: "4RK", "Boris", "Vay", "HC"`
  The numerical integration method used. `"4RK"` is the 4^{th}-order Runge-Kutta method, `"Boris"` is the Boris method, `"Vay"` is the Vay method, and `"HC"` is the Higuera-Cary method. For details on the integrator methods, see Ripperda et al. (2018). The Boris method has been validated for use in geomagnetic cut-off computations by Kruchinin et al. (2024) and is thus set as the default integrator.

- **gyropercent**
  `Type:` string
  `Default: "15"`
  Maximum time-step size as a percentage of the gyration period.

- **maxdistance**
  `Type:` float
  `Default: 100`
  Maximum distance that the particle can travel before computation termination [Earth radii].

- **maxtime**
  `Type:` float
  `Default: 0`
  Maximum time that can pass in the particle's frame of reference before computation is terminated [s]. maxtime = `0` means no time limit.

- **minaltitude**
  `Type:` float
  `Default: 20`
  Minimum altitude threshold before trajectory ends [km].

- **magnetopause**
  `Type:` string
  `Default: "Kobel"`
  `Options: "Sphere", "aFormisano", "Sibeck", "Kobel"`
  Model used to define the magnetopause boundary. `"Sphere"` uses a 25Re radii sphere, `"aFormisano"` uses the aberrated Formisano model Formisano et al. (1979), `"Sibeck"` uses the Sibeck model Sibeck et al. (1991), and `"Kobel"` uses the Kobel model Kobel (1992). When using Tsyganeko models later than the TSY89 model, a magnetopause is included within the code, and it will overwrite your selection for this variable.

3.8 Grid Options

- **maxlat, minlat, maxlong, minlong**
  `Type:` float
  `Default: 90,-90,360,0`
  When using the planet or trace function, users should define a grid of starting locations around the globe for the computations. `maxlat, minlat, maxlong, minlong` define the maximum and minimum values of the latitude and longitude. OTSO is set up so that the prime meridian (i.e. longitude = 0) passes through Greenwich, UK.

- **latstep, longstep**
  `Type:` float
  `Default: -5,5`
  The `latstep` and `longstep` arguments define the grid resolution for the computation. Default values mean that over the user-defined region (using `maxlat, minlat, maxlong, minlong`), a 5°x5° grid of latitude and longitude positions will be generated. Note that `latstep` is negative as the grid is generated from positive latitude to negative. If the user inputs a positive `latstep`, a warning will be thrown and the `latstep` will be converted to negative.

- **array_of_lats_and_longs**
  `Type:` list of list of floats `[[lat1,long1],[lat2,long2],...]`
  `Default: None`
  Users may enter their specific values of latitude and longitude to be used in the planet function, specifically if they wish not to generate a grid using the `maxlat, minlat, maxlong, minlong, latstep` and `longstep` arguments. If the user defines the `array_of_lats_and_longs` argument, it will overwrite the default grid generation parameters.

3.9 Asymptotic Conditions

- **asymptotic**
  `Type:` string
  `Default: "NO"`
  `Options: "NO" / "YES"`
  It can be used in the flight and planet functions. This will turn on asymptotic viewing direction computations for these functions in addition to the cut-off computations.

- **asymlevels**
  `Type:` list of floats
  `Default: [0.1,0.3,0.5,1,2,3,4,5,6,7,8,9,10,15,20,30,50,70,100,300,500,700,1000]`
  Defines the rigidity or energy levels for the particles to be tested when computing the asymptotic viewing directions using the `asymlevels` variable. The unit is either `GV` or `GeV`, and this is specified by using the `unit` variable.

- **unit**
  `Type:` string
  `Default: "GeV"`
  `Options: "GeV" / "GV"`
  Specifies the unit of the input `asymlevels` list argument.

3.10 Miscellaneous

- **coordsystem, Coordsys**
  `Type:` string
  `Default: "GEO", "GEO"`
  `Options: "GDZ","GEO","GSM","GSE","SM","GEI","MAG","SPH","RLL"`
  Coordinate system for the outputs of the simulation. `"GDZ"` is geodetic, `"GEO"` is geocentric, `"GSM"` is geocentric solar magnetospheric, `"GSE"` geocentric solar ecliptic, `"SM"` is solar magnetic, `"GEI"` is geocentric equatorial inertial, `"MAG"` is geomagnetic, `"SPH"` geocentric spherical, and `"RLL"` is regular latitude-longitude. NOTE: the asymptotic latitude and longitude can only be returned in `"GEO"` or `"GSE"` coordinate systems currently.

- **CoordIN, CoordOUT**
  `Type:` string
  `Default: None, None`
  `Options: "GDZ","GEO","GSM","GSE","SM","GEI","MAG","SPH","RLL"`
  Coordinate systems to be used in the coordinate transform function. Available coordinate systems are outlined in the `coordsystem` and `Coordsys` variables.

- **anti**
  `Type:` string
  `Default: "YES"`
  `Options: "YES" / "NO"`
  If `"YES"`, an antiparticle will be simulated. If `"NO"`, a normal particle will be simulated. e.g. `"NO"` = proton, and `"YES"` = anti-proton.

- **Anum**
  `Type:` integer
  `Default: 1`
  Sets the atomic number of the cosmic ray being simulated. `0` sets the CR to an electron, `1` is a proton, `2` is a helium ion and so on.

- **corenum**
  `Type:` integer
  `Default: 1`
  Number of CPU cores used. Will split the computations over the number of dedicated cores, for example, if computing computations for 4 locations and you enter `corenum = 2`, each core will deal with 2 computations, halving the computation time compared to using 1 core. OTSO will check to ensure that the input corenum doesn't exceed the actual number of cores on the user's computer.

- **serverdata**
  `Type:` string
  `Default: "OFF"`
  `Options: "ON" / "OFF"`
  The `serverdata` option allows OTSO to pull solar wind data and geomagnetic index data from the OMNI database for specific years (see https://spdf.gsfc.nasa.gov/pub/data/omni/). OTSO will use the OMNI data to compute the inputs needed for the Tsyganenko models, if able, and store all the data obtained locally for the input year value. Server data is stored within the OTSO package in CSV files named after the year the data is obtained from. `serverdata` is available back to 1963; however, due to the lack of measurements that far back OTSO is limited

to low-resolution OMNI data consisting mainly of geomagnetic index values until 1981.

- **livedata**
  `Type:` string
  `Default: "OFF"`
  `Options: "ON" / "OFF"`
  The `livedata` option allows OTSO to pull near-real-time solar wind and geomagnetic index data from online databases for the last 7 days (from the date of computation). Solar wind data is taken from the NOAA (see https://www.swpc.noaa.gov/products/real-time-solar-wind); Kp index data is taken from Gfz-Postdam (see https://kp.gfz.de/en/; and the Dst index is taken from the World Data Center for Geomagnetism (see https://wdc.kugi.kyoto-u.ac.jp/dstdir/). using `livedata` lets users use all Tsyganenko models except TSY04, which requires high-resolution OMNI data to compute the W1-W6 variables and is not updated quickly enough for real-time computations.

4 Functions

- Cutoff
- Cone
- Trajectory
- Planet
- Flight
- Trace
- Coordtrans
- Magfield

There are eight functions within OTSOpy, these being the computation of: cut-off rigidities, asymptotic cones, individual CR trajectories, planetary cut-off maps, flight path cut-offs, magnetic field line tracing, coordinate transforms, and magnetic field vector computations.

4.0.1 Cutoff

While OTSO repeats the same cosmic ray trajectories over the input range of rigidities, it can encounter a region known as the penumbra. Within this region, there is a mixture of allowed and forbidden cosmic ray trajectories, corresponding to being able to and not able to escape the magnetopause, respectively. To make sense of the penumbral region, an effective cut-off rigidity (R_c) is computed using the last accepted rigidity before the penumbra (R_U), the last accepted rigidity (R_L), and the size of the steps in rigidity over the range being tested (Δ). The equation for (R_c) is as follows:

$$R_c = R_U - \sum_{R_L}^{R_U} \Delta R_i(allowed) \tag{1}$$

Arguments:

- `Stations` (required)
- `customlocations`
- `startrigidity`, `endrigidity`, `rigiditystep`
- `rigidityscan`

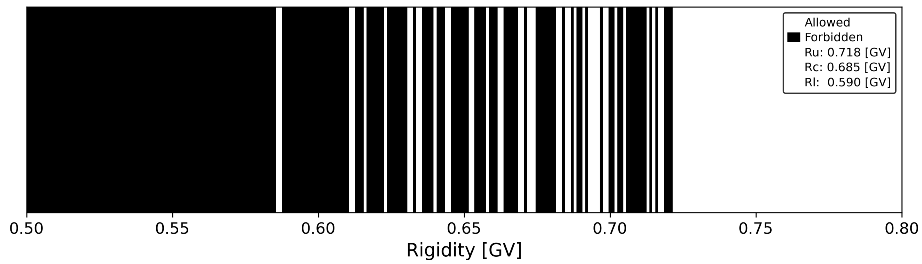

Figure 1: Computation of the Oulu neutron monitor effective cut-off rigidity using the IGRF 2000 epoch and TSY89 model with kp index = 0. Penumbra is shown by the forbidden and allowed trajectories being black and white, respectively. The upper and lower cut-off values (Ru and Rl) are denoted in the legend, from which the effective cut-off (Rc) is computed.

- `cutoff_comp`
- `zenith`, `azimuth`
- `startaltitude`, `minaltitude`
- `maxdistance`, `maxtime`
- `internalmag`, `externalmag`
- `serverdata`, `livedata`
- `vx, vy, vz, by, bz, density, pdyn`
- `kp, Dst, G1, G2, G3, W1, W2, W3, W4, W5, W6`
- `Anum, anti`
- `year, month, day, hour, minute, second`
- `intmodel, gyropercent`
- `coordsystem`
- `magnetopause`
- `corenum`
- `g, h`
- `MHDfile, MHDcoordsys`

4.1 Cone

The cone function conducts two key computations in tandem these being the asymptotic cones computation and effective cut-off rigidity calculations. Both of these results require the modelling of cosmic ray trajectories over a range of rigidity values that is defined by the user.

4.1.1 Asymptotic Cone

In order to determine the asymptotic cone of a point on the Earth (typically a neutron monitor location is selected) OTSO will compute the asymptotic latitude (Λ) and longitude (Ψ) of a CR once the modelling has been completed for that rigidity value. The equations for these values in the spherical coordinate system are:

$$\tan\Lambda = \frac{-v_\theta \sin\theta + v_r \cos\theta}{\sqrt{v_\varphi^2 + (v_\theta \cos\theta + v_r \sin\theta)^2}} \tag{2}$$

$$\Psi = \varphi + \arctan\left(\frac{v_\varphi}{v_\theta \cos\theta + v_r \sin\theta}\right) \tag{3}$$

where θ is the co-latitude and φ is the longitude. These calculations are done regardless of where the cosmic ray is at the end of the modelling.

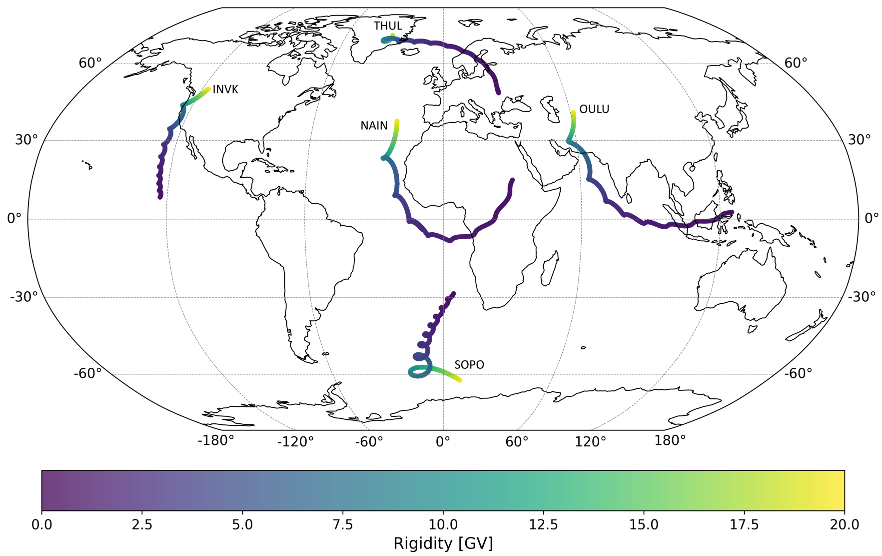

Figure 2: Asymptotic cones for the Oulu, Nain, South Pole, Thule, and Inuvik neutron monitors for the IGRF 2010 epoch and TSY89 model, with kp = 0. Latitudes and longitudes are in the geocentric coordinate system.

Arguments:

- `Stations` (required)
- `customlocations`
- `startrigidity`, `endrigidity`, `rigiditystep`
- `zenith`, `azimuth`
- `startaltitude`, `minaltitude`
- `maxdistance`, `maxtime`
- `internalmag`, `externalmag`
- `serverdata`, `livedata`
- `vx, vy, vz, by, bz, density, pdyn`
- `kp, Dst, G1, G2, G3, W1, W2, W3, W4, W5, W6`
- `Anum, anti`
- `year, month, day, hour, minute, second`
- `intmodel, gyropercent`
- `coordsystem`
- `magnetopause`
- `corenum`

- `g, h`

- `MHDfile, MHDcoordsys`

4.2 Trajectory

Invoking the trajectory function will make OTSO compute the path of a CR from a given point within the magnetosphere (typically 20km above the Earth's surface) using a numerical integration method selected by the user until one of three conditions is met. The conditions under which the computation stops are if the CR: encounters the model magnetopause, falls below an altitude of 20km, a certain amount of time passes in the CR's reference frame, or travels a distance of over 100 Earth radii (the minimum altitude, maximum time, and maximum distance travelled values can be edited). The trajectory will be considered allowed if the first condition is met and forbidden if either of the latter three is met.

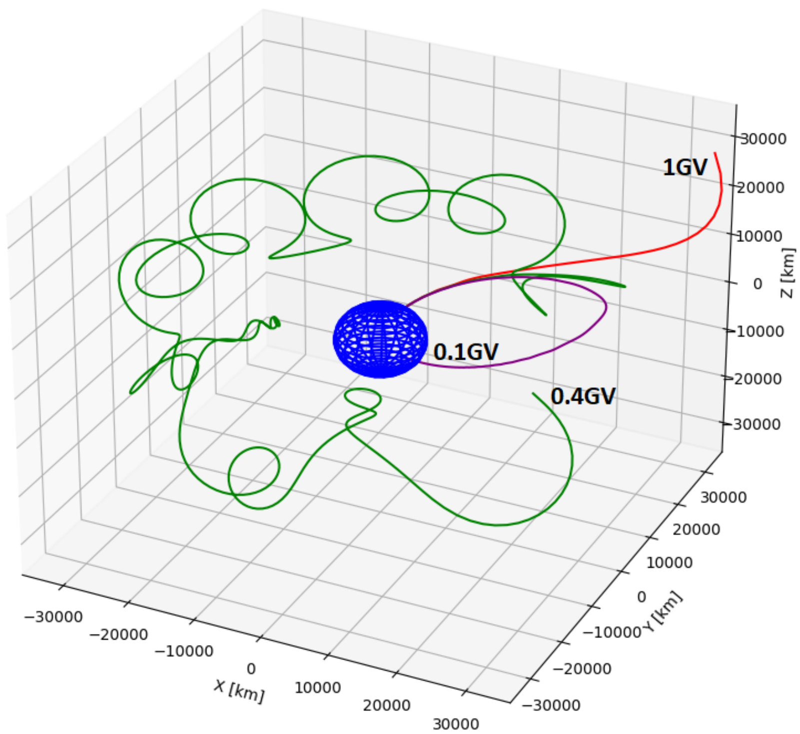

Figure 3: Computed trajectories of three cosmic rays of various rigidity values being backtraced from the Oulu neutron monitor for the IGRF 2000 and TSY89 model, with kp = 0. The 1GV particle is allowed (able to escape the magnetosphere); the 0.4GV particle is forbidden (it is trapped in the magnetosphere); and the 0.1GV is also forbidden (it returns to Earth).

Arguments:

- `Stations` (required)

- `rigidity` (required)

- `customlocations`

- `zenith`, `azimuth`

- `startaltitude`, `minaltitude`

- `maxdistance`, `maxtime`
- `internalmag`, `externalmag`
- `serverdata`, `livedata`
- `vx, vy, vz, by, bz, density, pdyn`
- `kp, Dst, G1, G2, G3, W1, W2, W3, W4, W5, W6`
- `Anum, anti`
- `year, month, day, hour, minute, second`
- `intmodel, gyropercent`
- `coordsystem`
- `magnetopause`
- `corenum`
- `g, h`
- `MHDfile, MHDcoordsys`

4.3 Planet

Invoking the planet function will perform the cutoff computation over a user-defined grid. This allows the generation of global cut-off maps.

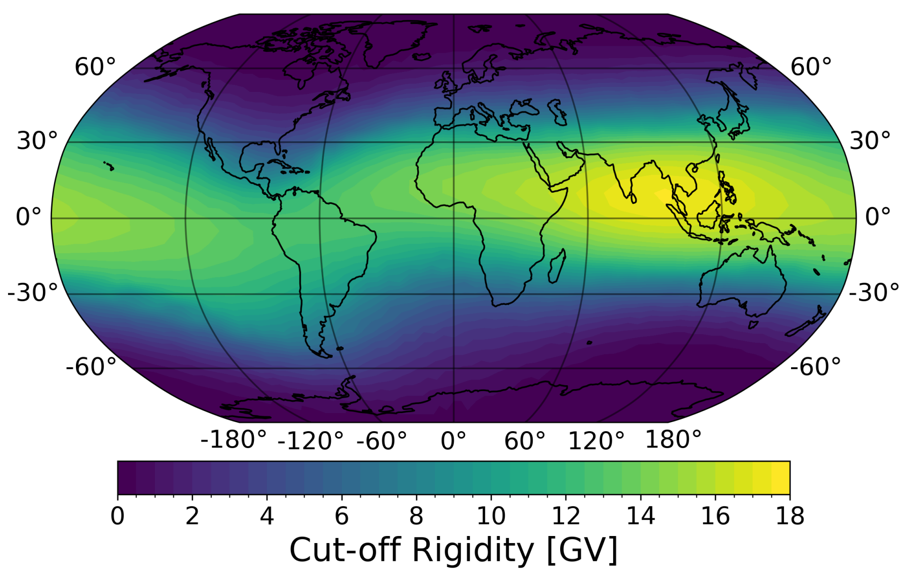

Figure 4: Computed vertical effective cut-off rigidities across a 5°x5° grid of the Earth. These computations were done using the IGRF 2000 epoch and TSY89 model, with kp = 0.

Arguments:

- `cutoff_comp`
- `startrigidity`, `endrigidity`, `rigiditystep`
- `rigidityscan`
- `startaltitude`, `minaltitude`

- maxdistance, maxtime
- internalmag, externalmag
- serverdata, livedata
- latstep, longstep
- maxlat, minlat, maxlong, minlong
- array_of_lats_and_longs
- vx, vy, vz, by, bz, density, pdyn
- kp, Dst, G1, G2, G3, W1, W2, W3, W4, W5, W6
- Anum, anti
- asymptotic, asymlevels, unit
- year, month, day, hour, minute, second
- intmodel, gyropercent
- magnetopause
- corenum
- g, h
- MHDfile, MHDcoordsys

4.4 Flight

Invoking the flight function will perform the cut-off computation over a user-defined path. Note this path doesn't need to be a flight path; it can also be the path of a ship for latitude surveys or low-Earth orbit spacecraft. Be careful that the altitude inputs are not below the minimum altitude, as this will lead to all trajectories being forbidden and erroneous cut-off computations.

Arguments:

- latitudes (required)
- longitudes (required)
- dates (required)
- altitudes (required)
- cutoff_comp
- startrigidity, endrigidity, rigiditystep
- rigidityscan
- startaltitude, minaltitude
- maxdistance, maxtime
- internalmag, externalmag
- serverdata, livedata
- vx, vy, vz, by, bz, density, pdyn
- kp, Dst, G1, G2, G3, W1, W2, W3, W4, W5, W6

- `Anum, anti`
- `asymptotic, asymlevels, unit`
- `intmodel, gyropercent`
- `magnetopause`
- `coordsystem`
- `corenum`
- `g, h`
- `MHDfile, MHDcoordsys`

4.5 Trace

Invoking the trace function will make OTSO trace the path of the magnetic field lines for the magnetic configuration that you have selected.

Arguments:

- `startaltitude`, `minaltitude`
- `maxdistance`, `maxtime`
- `internalmag`, `externalmag`
- `serverdata`, `livedata`
- `latstep`, `longstep`
- `maxlat`, `minlat`, `maxlong`, `minlong`
- `vx, vy, vz, by, bz, density, pdyn`
- `kp, Dst, G1, G2, G3, W1, W2, W3, W4, W5, W6`
- `Anum, anti`
- `year, month, day, hour, minute, second`
- `intmodel, gyropercent`
- `Coordsys`
- `magnetopause`
- `corenum`
- `g, h`
- `MHDfile, MHDcoordsys`

4.6 Coordtrans

Transforms input positional data from one coordinate system to another.

Arguments:

- `Locations` (required)
- `dates` (required)
- `CoordIN`, `CoordOUT` (required)
- `corenum`

4.7 Magfield

The magfield function will return the magnetic field vector at a given location in the Earth's magnetosphere.

Arguments:

- `Locations` (required)
- `internalmag`, `externalmag`
- `serverdata`, `livedata`
- `vx, vy, vz, by, bz, density, pdyn`
- `kp, Dst, G1, G2, G3, W1, W2, W3, W4, W5, W6`
- `year, month, day, hour, minute, second`
- `coordsystem`
- `corenum`
- `g, h`
- `MHDfile, MHDcoordsys`

5 How to Use

Once downloaded, the user must import the OTSO package using:

```
import OTSO
```

As multicore processing is inherent in OTSO it is important that the code that utilises OTSO must be nested in a main if statement:

```
if __name__ == '__main__':
```

Any OTSO function can then be called. See below for an example evoking the cutoff function.

```
import OTSO

if __name__ == '__main__':

    stations_list = ["OULU","ROME","ATHN","CALG"] # list of neutron monitor stations (using
        their abbreviations)

    cutoff = OTSO.cutoff(Stations=stations_list,corenum=1,year=2000,month=1,day=1,hour=0)


    print(cutoff[0]) # dataframe output containing Ru, Rc, Rl for all input locations
    print(cutoff[1]) # text output of input variable information
```

Further examples can be found on the GitHub (https://github.com/NLarsen15/OTSOpy) page and within the README.md file.

5.1 Output Text

Every OTSO function returns an array. The last value within the array is always text that details the specifics of the computation that was done, such as input variables and date of computation, etc. It is recommended that users save this text to a .txt file when using OTSO for reference later.

Users can access this text reliably by using:

```
text = output_array[-1]
```

An example of the output text can be seen below for the cutoff function.

```
Date of OTSO computation: 2025-05-03
Total computation time: 11.988 seconds

Cutoff Computed: Vertical Cutoff Rigidity

Output Coordinate System:
GEO

Rigidity Scan:
Rigidity Scan Used

Integration Method:
Boris Method

Input Variables:

Data Used: User Inputted Data Used

Simulation Date: 01/01/2000, 00:00:00

Max Time Step [% of gyrofrequency]: 15.0

Minimum Altitude: 20km
Max Distance: 100Re
Max Time: 0s

Start Altitude = 20km
Zenith = 0
Azimuth = 0

Kp = 0
IOPT = 1

Solar Wind Speed [km/s]:
Vx = 500
Vy = 0
Vz = 0

IMF [nT]:
By = 5
Bz = 5

Density = 1 cm^-3
Pdyn = 0.418 nPa

Dst = 0 nT

G1 = 0
G2 = 0
G3 = 0
```

```
W1 = 0
W2 = 0
W3 = 0
W4 = 0
W5 = 0
W6 = 0

Atomic Number = 1

Particle Type = anti-particle

Magnetic Field Models:
Internal Model = IGRF
External Model = Tsyganenko 89

Magnetopause Model = Kobel Model

Rigidity:
Start = 20 [GV]
End = 0 [GV]
Step = 0.01 [GV]

Stations:
ATHN, Latitude: 37.58, Longitude: 23.47
CALG, Latitude: 51.08, Longitude: -114.13
OULU, Latitude: 65.05, Longitude: 25.47
ROME, Latitude: 41.86, Longitude: 12.47
```

References

Bieber, J.W., Clem, J., Evenson, P., 1997. Efficient Computation of Apparent Cutoffs, in: International Cosmic Ray Conference, p. 389.

Boberg, P.R., Tylka, A.J., Adams Jr., J.H., Flückiger, E.O., Kobel, E., 1995. Geomagnetic transmission of solar energetic protons during the geomagnetic disturbances of october 1989. Geophysical Research Letters 22, 1133–1136. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/95GL00948, doi:https://doi.org/10.1029/95GL00948, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/95GL00948.

Formisano, V., Domingo, V., Wenzel, K.P., 1979. The three-dimensional shape of the magnetopause. Planetary and Space Science 27, 1137–1149. URL: https://www.sciencedirect.com/science/article/pii/003206337990134X, doi:https://doi.org/10.1016/0032-0633(79)90134-X.

Kobel, E., 1992. Zu den magnetosphärischen Effekten der kosmischen Strahlung. Inaugural dissertation. University of Bern.

Kruchinin, P.A., Malakhov, V.V., Golubkov, V.S., Mayorov, A.G., 2024. Calculation of Geomagnetic Cutoff Rigidity Using Tracing Based on the Buneman–Boris Method. Geomagnetism and Aeronomy 64, 735–742. URL: https://doi.org/10.1134/S0016793224600668, doi:10.1134/S0016793224600668.

Larsen, N., Mishev, A., Usoskin, I., 2023. A new open-source geomagnetosphere propagation tool (otso) and its applications. Journal of Geophysical Research: Space Physics 128, e2022JA031061. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2022JA031061, doi:https://doi.org/10.1029/2022JA031061, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2022JA031061. e2022JA031061 2022JA031061.

Ripperda, B., Bacchini, F., Teunissen, J., Xia, C., Porth, O., Sironi, L., Lapenta, G., Keppens, R., 2018. A comprehensive comparison of relativistic particle integrators. The Astrophysical Journal

Supplement Series 235, 21. URL: https://dx.doi.org/10.3847/1538-4365/aab114, doi:10.3847/1538-4365/aab114.

Sibeck, D.G., Lopez, R.E., Roelof, E.C., 1991. Solar wind control of the magnetopause shape, location, and motion. Journal of Geophysical Research 96, 5489–5495. URL: http://inis.iaea.org/search/search.aspx?orig_q=RN:22089274.

Tsyganenko, N., 1987. Global quantitative models of the geomagnetic field in the cislunar magnetosphere for different disturbance levels. Planetary and Space Science 35, 1347–1358. URL: https://www.sciencedirect.com/science/article/pii/0032063387900468, doi:https://doi.org/10.1016/0032-0633(87)90046-8.

Tsyganenko, N., 1989. A magnetospheric magnetic field model with a warped tail current sheet. Planetary and Space Science 37, 5–20.

Tsyganenko, N., 1995. Modeling the earth's magnetospheric magnetic field confined within a realistic magnetopause. Journal of Geophysical Research:Space Physics 100, 5599–5612. doi:110.1029/94JA03193.

Tsyganenko, N., Singer, H., Kasper, J., 2003. Storm-time distortion of the inner magnetosphere: How severe can it get. Journal of Geophysical Research 108. doi:10.1029/2002JA009808.

Tsyganenko, N.A., 2002a. A model of the near magnetosphere with a dawn-dusk asymmetry 1. mathematical structure. Journal of Geophysical Research: Space Physics 107, SMP 12–1–SMP 12–15. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2001JA000219, doi:https://doi.org/10.1029/2001JA000219, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2001JA000219.

Tsyganenko, N.A., 2002b. A model of the near magnetosphere with a dawn-dusk asymmetry 2. parameterization and fitting to observations. Journal of Geophysical Research: Space Physics 107, SMP 10–1–SMP 10–17. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2001JA000220, doi:https://doi.org/10.1029/2001JA000220, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2001JA000220.

Tsyganenko, N.A., Sitnov, M.I., 2005. Modeling the dynamics of the inner magnetosphere during strong geomagnetic storms. Journal of Geophysical Research: Space Physics 110. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2004JA010798, doi:https://doi.org/10.1029/2004JA010798.