
ProcessMCRaT

Tyler Parsotan

Apr 07, 2021

CONTENTS:

1	mclib.py	1
2	read_process_files.py	7
3	mcrat_movie.py	19
4	Indices and tables	23
	Python Module Index	25
	Index	27

MCLIB.PY

`mclib.Lorentz_Boost` (*boost*, *P_ph*)

This routine performs a Lorentz boost of a 4-momentum. The boost is specified with a 3-vel.

Parameters

- **boost** – The velocity vector of the frame that the photon will be boosted into.
- **P_ph** – The photon 4 momentum that will be boosted into the desired frame of reference.

Returns the boosted photon 4 mometum

`mclib.event3D` (*r_obs*, *theta_deg*, *phi_deg*, *dtheta_deg*, *path*, *lastfile*, *sim_type*, *riken_switch=False*)

Place holder function to conduct a synthetic observation of MCRaT simulated outflow using a 3D hydro simulationp.

Parameters

- **r_obs** –
- **theta_deg** –
- **phi_deg** –
- **dtheta_deg** –
- **path** –
- **lastfile** –
- **sim_type** –
- **riken_switch** –

Returns

`mclib.event4` (*r_obs*, *theta_deg*, *dtheta_deg*, *path*, *lastfile*, *sim_type*, *withintheta_deg=False*,
riken_switch=False, *save_event=True*)

Function to collect MCRaT photons in space that will be observed. legacy code from when MCRaT produced text files as its output.

Parameters

- **r_obs** – radius of where the detector will be placed, should be at a radius such that all the photons have propagated past the detector by the end of the MCRaT simulation
- **theta_deg** – observer viewing angle in degrees
- **dtheta_deg** – delta theta of the observer viewieng angle for accepting photons, also in degrees
- **path** – path to the directory that holds ll the output MCRaT files, should be a string

- **lastfile** – the number of the last MCRaT output file, this should be an int
- **sim_type** – A string that will be the name of the output file of this function
- **withintheta_deg** – this is a depreciated key
- **riken_switch** – this is a depreciated key
- **save_event** – switch to create the event file that will be read in by other functions

Returns returns photon's that would be observed and thier energy, detector crossing time, and their weight

```
mclib.event_h5(r_obs, theta_deg, dtheta_deg, path, lastfile, sim_type, fps=5, read_comv=False,
               read_stokes=False, read_type=False, save_event_file=True, append=False)
```

Function to collect MCRaT photons in space that will be observed. Saves event files in EVENT_FILES/ directory that must be created before calling this functionp.

Parameters

- **r_obs** – radius of where the detector will be placed, should be at a radius such that all the photons have propagated past the detector by the end of the MCRaT simulation
- **theta_deg** – observer viewing angle in degrees
- **dtheta_deg** – delta theta of the observer viewing angle for accepting photons, also in degrees
- **path** – path to the directory that holds ll the output MCRaT files, should be a string
- **lastfile** – the number of the last MCRaT output file, this should be an int
- **sim_type** – A string that will be the name of the output file of this function
- **fps** – an int that represents the number fo frames per second in the hydro simulation used
- **read_comv** – switch to denote of the MCRaT files contain the comoving photon 4 momenta
- **read_stokes** – switch to denote if the MCRaT output files have the stokes parameters
- **save_event_file** – switch to determine if the function should save the event file or not
- **append** – switch to denote if the function should append the extracted data to a pre-existing event file

Returns returns (N,8) array where N is the number of photons 'detected'. The array contains: the crossing time of the photon, the lab energy in keV, the weight, the index in the list of photons, the stokes parameters and the comoving energy in keV

```
mclib.lcur(simid, t, units='erg/s', theta=1.0, dtheta=1.0, phi=0, dphi=1, sim_dims=2, h5=True, photon_type=None, energy_range=None)
```

reads in the event file and bins photons in uniform time bins to create light curves

Parameters

- **simid** – a string of the event file base name of the event file created in event_h5 (everything but the .evt)
- **t** – an array of time bin edges
- **units** – string specifying units, can be erg/s or cts/s
- **theta** – observer viewing angle in degrees
- **dtheta** – the size of the observer viewing angle bin in degrees (same as is specified in event_h5 function)

- **phi** – azimuthal angle for mock observer in degrees (only for 3D simulations, not fully supported)
- **dphi** – the size of the observer azimuthal viewing angle bin in degrees (only for 3D simulations, not fully supported)
- **sim_dims** – The number of dimensions of the hydro simulation used
- **h5** – specify if the format of the MCRaT output files is hdf5 files or not (denotes if using an old format or a newer format of saving files)
- **photon_type** – can be set to 's', 'i', or left as None in order to select thermal synchrotron photons, injected photons, or all the photons in the simulation for analysis
- **energy_range** – has units of keV, can be left as None to choose photons of all energy ranges for analysis or it can be set to an array with [min energy, max energy] e.g. [1, 10] for 1 to 10 keV (limits inclusive)

Returns returns the time binned quantities of luminosity, luminosity error, number of photons in each bin, the average energy of photons in each bin, polarization, the stokes parameters, polarization error and polarization angle

`mclib.lcur_old(simid, t, units='erg/s', theta=1.0, dtheta=1.0, phi=0, dphi=1, sim_dims=2, h5=True, photon_type=None, energy_range=None)`
reads in the event file and bins photons in uniform time bins to create light curves

Parameters

- **simid** – a string of the event file base name of the event file created in event_h5 (everything but the .evt)
- **t** – an array of time bin edges
- **units** – string specifying units, can be erg/s or cts/s
- **theta** – observer viewing angle in degrees
- **dtheta** – the size of the observer viewing angle bin in degrees (same as is specified in event_h5 function)
- **phi** – azimuthal angle for mock observer in degrees (only for 3D simulations, not fully supported)
- **dphi** – the size of the observer azimuthal viewing angle bin in degrees (only for 3D simulations, not fully supported)
- **sim_dims** – The number of dimensions of the hydro simulation used
- **h5** – specify if the format of the MCRaT output files is hdf5 files or not (denotes if using an old format or a newer format of saving files)
- **photon_type** – can be set to 's', 'i', or left as None in order to select thermal synchrotron photons, injected photons, or all the photons in the simulation for analysis
- **energy_range** – has units of keV, can be left as None to choose photons of all energy ranges for analysis or it can be set to an array with [min energy, max energy] e.g. [1, 10] for 1 to 10 keV (limits inclusive)

Returns returns the time binned quantities of luminosity, luminosity error, number of photons in each bin, the average energy of photons in each bin, polarization, the stokes parameters, polarization error and polarization angle

`mclib.lcur_var_t` (*simid, time_start, time_end, dt, dt_min=0.2, liso_c=1e+50, units='erg/s', theta=1.0, dtheta=1.0, phi=0, dphi=1, sim_dims=2, h5=True, photon_type=None, energy_range=None, use_Lcrit=False*)

Produces time binned quantities for non-uniform time bins. The time bins must be larger than some critical luminosity and some minimum dt that the user specifies.

Parameters

- **simid** – a string of the event file base name of the event file created in event_h5 (everything but the .evt)
- **time_start** – starting time of the light curve
- **time_end** – end time of the light curve binning
- **dt** – initial dt of the time bins
- **dt_min** – the minimum acceptable dt for the light curve
- **liso_c** – the minimum isotropic luminosity for a given time bin (in the same units specified by units)
- **units** – a string of the units of the light curve that will be produced (erg/s or cts/s)
- **theta** – the observer viewing angle in degrees
- **dtheta** – the size of the observer viewing angle bin in degrees (same as is specified in event_h5 function)
- **phi** – azimuthal angle for mock observer in degrees (only for 3D simulations, not fully supported)
- **dphi** – the size of the observer azimuthal viewing angle bin in degrees (only for 3D simulations, not fully supported)
- **sim_dims** – The number of dimensions of the hydro simulation used
- **h5** – specify if the format of the MCRaT output files is hdf5 files or not (denotes if using an old format or a newer format of saving files)
- **photon_type** – can be set to 's', 'i', or left as None in order to select thermal synchrotron photons, injected photons, or all the photons in the simulation for analysis
- **energy_range** – has units of keV, can be left as None to choose photons of all energy ranges for analysis or it can be set to an array with [min energy, max energy] e.g. [1, 10] for 1 to 10 keV (limits inclusive)

Returns returns the time binned quantities of luminosity, luminosity error, number of photons in each bin, the average energy of photons in each bin, polarization, the stokes parameters, polarization error and polarization angle

`mclib.lorentzBoostVectorized` (*boost, P_ph*)

Function to quickly lorentz boost a set of photon 4 momenta, and ensures that the 0 norm condition is met.

Parameters

- **boost** – The velocity vector of the frame that the photon will be boosted into. The shape of the array should be (3,N), where N is the number of photons that will be lorentz boosted
- **P_ph** – The photon's 4 momentum that will be boosted into the desired frame of reference. The shape of the array should be (4,N)

Returns returns (4,N) array of the boosted photon's 4 momenta

`mclib.read_event_file` (*simid, h5=True*)

An example function for reading the event files produced by event_h5.

Parameters

- **simid** – the name of the event file to be read
- **h5** – switch to set whether the MCRaT simulation used hdf5 files (should always be set to true for the current versions of MCRaT)

Returns returns the data from the event file

`mclib.readanddecimate(fnam, inj_radius)`

Legacy code from the python version of MCRaT to read and process FLASH files

Parameters

- **fnam** – string of directory and file name of the FLASH file that will be loaded and processed
- **inj_radius** – The radius that photons are injected in cm

Returns returns FLASH values of various quantities at cells

`mclib.single_cs(P_el, P_ph)`

This function conducts a single compton scatter in the electron rest frame. The photon has to be in the comoving frame first. Legacy code from Python version of MCRaT.

Parameters

- **P_el** – electron 4 momentum
- **P_ph** – photon comoving 4 momentum

Returns returns post-scattered photon 4 momentum in the comoving frame

`mclib.single_electron(T, P_ph)`

This routine simulates one electron from a thermal distribution at a temperature T. The electron has the correct angular distribution for a photon traveling with direction P_ph

`mclib.spex(simid, numin, numax, tmin, tmax, units='erg/s', h5=True, photon_type=None, calc_pol=False)`

Produces spectra of photons detected within any time interval

Parameters

- **simid** – a string of the event file base name of the event file created in event_h5 (everything but the .evt)
- **numin** – array of energy values of the left most cutoff of the energy bins in the spectrum in keV
- **numax** – array of energy values of the right most cutoff of the energy bins in the spectrum in keV
- **tmin** – minimum of time bin that we are interested in collecting photons in to analyze
- **tmax** – max of time bin that we are interested in collecting photons in to analyze
- **units** – a string of the units of the spectrum that will be produced (erg/s or cts/s), each unit is then divided by the width of the energy bin in keV
- **h5** – specify if the format of the MCRaT output files is hdf5 files or not (denotes if using an old format or a newer format of saving files)
- **photon_type** – can be set to 's', 'i', or left as None in order to select thermal synchrotron photons, injected photons, or all the photons in the simulation for analysis
- **energy_range** – has units of keV, can be left as None to choose photons of all energy ranges for analysis or it can be set to an array with [min energy, max energy] e.g. [1, 10] for 1 to 10 keV (limits inclusive)

Returns returns the spectrum with the specified units, for the photons in each energy bin

`mclib.zero_norm(P)`

Takes a photon 4 momentum and checks if it satisfies the 0 norm condition of photons. If not, it corrects the 4 momentum by assuming that the energy is correct.

Parameters P – photon 4 momentum

Returns returns the correct 0 normed photon 4 momentum

READ_PROCESS_FILES.PY

`read_process_files.Band(x, a, b, c, d)`

Function that evaluates the Band function

Parameters

- **x** – x is the energy in keV
- **a** – alpha parameter
- **b** – beta parameter
- **c** – break energy in keV
- **d** – normalization of the returned spectrum

Returns Returns the Band function evaluated for the parameters of interest (units of cts/s)

`read_process_files.Goodman(max_spex, max_energy)`

Function that returns Goodman's scalable spherical explosion spectra to compare against a spectra acquired by a spherical explosion run in MCRAT. To compare this to simulation data, the simulation spectrum needs to be in units of erg/s.

Parameters

- **max_spex** – maximum value of the data's spectrum.
- **max_energy** – The energy bin where the maximum of the data lies

Returns returns the scaled version of Goodman's spectrum

`read_process_files.amati(value='o')`

Return the Amati relationship and it's 1 sigma dispersion as given by Tsutsui et al. (2009).

Parameters **value** – a string that can be 'o', '+', or '-'. The default is set to 'o' for the actual Amati relationship. '+' gives the upper bound of uncertainty and '-' gives the lower bound of uncertainty.

Returns returns arrays of the a and y values of the amati relation/ error in the relation

`read_process_files.band_hist_data(pickle_file, event_files, time_start, time_end, dt, plotting=False, save_plot=False, choose_best=False)`

Function that collects a set of parameters from time resolved spectral fits of MCRaT data. This allows them to be histogrammed to look at the distribution of alpha, beta and e_pk for example.

Parameters

- **pickle_file** – the name of the pickle file that will contain all of the spectral parameter values so it is only necessary to run this lengthy calculation once.
- **event_files** – The event files that will be analyzed to fit time resolved spectra and acquire the spectral parameters, a 1D array of strings

- **time_start** – The start time of the light curve that will be analyzed, a 1D array that is the same size as event_files
- **time_end** – The end time of the light curve that will be analyzed, a 1D array that is the same size as event_files
- **dt** – The width of time bins that photons will be collected within to produce the time resolved spectra, this will be across all of the event files that will be analyzed
- **plotting** – Switch to determine if the call to produce the light curves for each event file should plot all the light curves that it produces so the user can inspect the spectral fits in each time interval
- **save_plot** – Switch to determine if the call to produce the light curves for each event file should save all the light curves that it produces (saves time in that the user doesn't have to recall the lcure_param function to make plots of the light curves)
- **choose_best** – Switch to ignore spectral parameters that haven't been well constrained (they may not have error bars)

Returns Returns 2D arrays of alpha, beta, and break energies that contain all the time resolved parameter fits for each event file that was passed to the function.

`read_process_files.band_hist_plot(pickle_file, plotting=True, save_plot=False, scale=1)`

Function to plot the histogrammed Band/COMP spectral parameters.

Parameters

- **pickle_file** – The produced pickle file name from the band_hist_data function
- **plotting** – Switch to show the produced plots
- **save_plot** – Switch to save the plotted histograms
- **scale** – Scaling to scale the histograms if necessary

Returns Returns 2D arrays of alpha, beta, and break energies that contain all the time resolved parameter fits for each event file that was passed to the band_hist_data function.

`read_process_files.blackbody(x, t, d)`

Function that produces a blackbody spectrum.

Parameters

- **x** – Energy in keV
- **t** – temp in kelvin
- **d** – normalization of the spectrum

Returns returns the blackbody spectrum parameterized by some temperature at the energies of interest

`read_process_files.cfit(event_file, time_start, time_end, dnulog=0.1, hdf5=True, plotting=False, save_plot=False, save_spe=None, append=False, photon_type=None, calc_pol=False, calc_synch=False)`

Function that fits either the COMP function of the Band function to time integrated or time resolved spectra from MCRaT and also acquires the 1 sigma error in the fitted values. The best fit spectra, accounting for the extra degree of freedom is chosen by conducting an F-test.

Parameters

- **event_file** – the base name of the event file that holds all of the info about the mock observed photons from MCRaT
- **time_start** – the start time of a given time bin to collect photons into a spectrum

- **time_end** – the end of the time bin for collecting photons into a spectrum to be fit by this function
- **dnulog** – the spacing between energy bins when binning the mock observed MCRaT photons (in keV)
- **hdf5** – switch to let the function know if the MCRaT output files are hdf5 files or not
- **plotting** – switch to let the function know that it should plot the MCRaT spectra and the fits to the spectra
- **save_plot** – switch to let the function know if it should save the plots
- **save_spex** – string to let the function save the tabulated spectrum to a file given by this string
- **append** – switch to let the function know if it should append data to the file given by save_spex
- **photon_type** – can be set to 's', 'i', or left as None in order to select thermal synchrotron photons, injected photons, or all the photons in the simulation for analysis
- **calc_pol** – switch to determine if polarization as a function of energy should be calculated and plotted

Returns returns the best fitted parameters to the function, a (3,2) array that has each parameters' (alpha, beta and break energy) upper and lower 1 sigma errors, and a character denoting the function that provided the best fit ('c' for COMP or 'b' for Band)

`read_process_files.comp(x, a, c, d)`

returns the Comptonized (COMP) spectrum for a given set of energies

Parameters

- **x** – the energies that the Comptonized function will be evaluated at (should be in keV)
- **a** – the alpha parameter
- **c** – the break energy
- **d** – the normalization of the spectrum

Returns returns the COMP spectrum at the energies of interest (units of cts/s)

`read_process_files.fluidGammaVsTheta(fluid_dir, r_interest, t_interest, fps, theta_max)`

Function to calculate the lorentz factor profile as a function of theta directly from the hydrodynamic files. It calculates which frame to analyze based off of what radius the user is interested in and the times in the outflow the user is interested in

Parameters

- **fluid_dir** – String that points to the directory of the hydrodynamic files, should include the base name of the hydro file (excluding any numbers)
- **r_interest** – The radius at which the user wants the flow to be analyzed
- **t_interest** – An array of times in the outflow that the user wants to be analyzed
- **fps** – The frames per second of the passed hydrodynamic simulation
- **theta_max** – The maximum angle to get the lorentz factor profile

Returns All the theta data points, in degrees, the average lorentz profile at the angle data points, and the array of the times that the user was interested in

`read_process_files.get_FERMI_best_data()`

A function to acquire data about the FERMI Best GRB sample, as is saved in the file named FERMI_BEST_GRB.dat. The data is from Yu et al. (2016).

Returns returns arrays of the Band or COMP function fitted GRB spectral parameters

`read_process_files.get_amati_rel(simid_array, time_start, time_end, save_plot=False, h5=False)`

A function to plot the MCRaT analyzed GRB alongside the amati relationship and a list of observed GRBs as provided by GRB_list.dat (observed GRBS are from Nava et al. (2012))

Parameters

- **simid_array** – An array of the event file name bases of the MCRaT result that will be plotted
- **t_start** – An array of the start times for each event file passed to simid_array
- **t_end** – An array of the end times for each event file passed to simid_array
- **save_plot** – Switch to save the plot as a PDF or not
- **h5** – Switch to specify if the MCRaT output files were HDF5 format or not, default is True. Should be set to False when an old version of MCRaT was used which outputted the data in text files

Returns No returns

`read_process_files.get_lundman_pol_data(p=4, thetaj=0.01)`

This function gets the appropriate values of polarization vs theta for various values of p and theta_j from the plots of Lundman et al. (2014b). The only accepted parameter combinations are p=4 with thetaj=0.01 or 0.1 and p=2 with thetaj=0.01

Parameters

- **p** – The p value for the width of the analytic jet from Lundman et al. (2014b)
- **thetaj** – The opening angle of the analytic jet profile from Lundman et al. (2014b)

Returns Returns a (nx2) array where n is the number of angles points read in from the file and the columns are the theta/thetaj values followed by the polarization degree

`read_process_files.get_yonetoku_rel(simid_array, t_start, t_end, dt=1, h5=False, save_plot=False)`

A function to plot the MCRaT analyzed GRB alongside the yonetoku relationship and a list of observed GRBs as provided by GRB_list.dat (observed GRBS are from Nava et al. (2012))

Parameters

- **simid_array** – An array of the event file name bases of the MCRaT result that will be plotted
- **t_start** – An array of the start times for each event file passed to simid_array
- **t_end** – An array of the end times for each event file passed to simid_array
- **dt** – The time bin width, default is 1 following how the relationship is derived
- **h5** – Switch to specify if the MCRaT output files were HDF5 format or not, default is True. Should be set to False when an old version of MCRaT was used which outputted the data in text files
- **save_plot** – Switch to save the plot as a PDF or not

Returns Returns arrays of the plotted peak energy, its error, the luminosity and the error in the luminosity for each event file that was provided to simid_array

`read_process_files.golenetskii` (*value='o'*)

Return the golenetskii relationship and its 2 sigma dispersion as given by Lu et al. (2012).

Parameters **value** – a string that can be ‘o’, ‘+’, or ‘-’. The default is set to ‘o’ for the actual golenetskii relationship. ‘+’ gives the upper bound of uncertainty and ‘-’ gives the lower bound of uncertainty.

Returns returns arrays of the a and y values of the relation/ error in the relation

`read_process_files.lcur_param` (*event_file, time_start, time_end, dt=1, phi_angle=0, delta_theta=1, unit='erg/s', plotting=True, save_plot=False, choose_best=False, lc_only=False, pol_only=False, dim=2, hdf5=True, save_spex=None, photon_type=None, energy_range=None*)

This function takes the MCRaT ‘detected’ photons from the event file and time bins them to produce light curves. The photons in a given time bin are then fitted with either the COMP or Band functions.

Parameters

- **event_file** – the base name of the event file that holds all of the info about the mock observed photons from MCRaT
- **time_start** – the start time to begin binning the photons
- **time_end** – the end time to stop binning the photons
- **dt** – the light curve time bin width
- **phi_angle** – unused parameter, meant for future 3D development
- **delta_theta** – delta theta of the observer viewing angle for accepting photons, in degrees. Should be the same value as what is passed into event_h5’s dtheta_deg
- **unit** – A string of the units that will be used for the light curve/spectrum ‘erg/s’ or ‘cts/s’
- **plotting** – Switch to allow the function to show the plots of the light curve/spectral parameters
- **save_plot** – Switch to allow the function to save the produced plot
- **choose_best** – Switch that permits the function to overwrite spectral parameters that do not have error bars to be ignored if set to True. If this is set to False all spectral parameters will be considered/plotted
- **lc_only** – switch that tells the function to only plot the light curve and peak energy of the time resolved spectra and none of the other parameters
- **pol_only** – Switch to allow the function to only plot the light curve and the polarization when set to True
- **dim** – The number of dimensions of the MCRaT simulation that is being analyzed. There is no support for 3D at this time so this switch should be set to 2, which is its default value
- **hdf5** – Switch to specify if the MCRaT output files were HDF5 format or not, default is True. Should be set to False when an old version of MCRaT was used which outputted the data in text files
- **save_spex** – Switch that allows the time resolved spectra to be saved to a text file
- **photon_type** – can be set to ‘s’, ‘i’, or left as None in order to select thermal synchrotron photons, injected photons, or all the photons in the simulation for analysis
- **energy_range** – has units of keV, can be left as None to choose photons of all energy ranges for analysis or it can be set to an array with [min energy, max energy] e.g. [1, 10] for 1 to 10 keV (limits inclusive)

Returns returns arrays of the various time resolved parameters and their errors. The order of the variables returned are: light curve, light curve error, alpha,beta,break energy, errors in the spectral parameters (a (times.size,3,2) array where times is the start of the light curve time bins), start times of the time bins, the best fit model in each time bin, polarization, stokes I, stokes Q, stokes U, stokes V, polarization and polarization angle errors (a ((times.size,2)) array), polarization angle, number of photons in each time bin

```
read_process_files.lcur_param_plot(event_file, lcur, lcur_e, alpha, beta, e_o, err, t, model, P, I,  
                                   Q, U, V, Perr, P_angle, num_scatt, dt=1, lc_only=False,  
                                   pol_only=False, plotting=True, h5=False, liso_c=None,  
                                   plot_optical=True)
```

The plotting function for the results of lcur_param and lcur_param_var_t functions.

Parameters

- **event_file** – The event file name, a string that will be incorporated into the saved pdf of the plot that is produced
- **lcur** – the values of the light curve at each time bin as an array
- **lcur_e** – the values of the error in the light curve values at each time bin
- **alpha** – an array of the fitted alpha parameters for each time resolved spectrum
- **beta** – an array of the fitted beta parameters for each time resolved spectrum
- **e_o** – an array of the fitted break energies for each time resolved spectrum
- **err** – A (t.size,3,2) array of errors in the spectral parameters
- **t** – An array of time bins both beginning and end that will be plotted
- **model** – An array of model types for the types of spectrum that best fit each time binned spectrum
- **P** – An array of the polarization at each time bin
- **I** –
- **Q** –
- **U** –
- **V** –
- **Perr** – The error in the polarization at each time bin
- **P_angle** – The polarization angle of the photons detected in each time bin
- **dt** – the size of the time bin if the time bins are uniformly size otherwise set it to be a negative number
- **lc_only** – Switch to denote if the function should only plot the light curve or not
- **pol_only** – Switch to denote if the function should plot just the light curve and the polarization parameters
- **plotting** – Switch, set to True by default, to show the plot produced
- **h5** – A switch to denote that the MCRaT simulation that is being analyzed produced HDF5 files (for backwards compatibility)
- **liso_c** – The value of the critical isotropic luminosity that was used to create the variable time bins

Returns No returns

```
read_process_files.lcur_param_var_t(event_file, time_start, time_end, dt=0.2, dt_min=0.2,
                                     phi_angle=0, delta_theta=1, liso_c=None, unit='erg/s',
                                     plotting=True, save_plot=False, choose_best=False,
                                     lc_only=False, pol_only=False, dim=2,
                                     riken_switch=False, hdf5=True, save_spex=None, photon_type=None,
                                     energy_range=None, use_Lcrit=False, plot3=False)
```

This function takes the MCRaT ‘detected’ photons from the event file and time bins them to produce light curves with variable time bins. The photons in a given time bin are then fitted with either the COMP or Band functions.

This function should be called after viewing the data in uniform time bins. It uses a minimum luminosity cutoff to form time bins, if this critical limit is achieved, the time bin is recorded otherwise the function will keep increasing a giving time bin in a light curve to ensure that the critical luminosity is achieved.

Parameters

- **event_file** – the base name of the event file that holds all of the info about the mock observed photons from MCRaT
- **time** – a python list or numpy array with the start time and the end time in chronological order e.g. time=[0, 35]
- **dt** – the light curve time bin width
- **phi_angle** – unused parameter, meant for future 3D development
- **delta_theta** – delta theta of the observer viewing angle for accepting photons, in degrees. Should be the same value as what is passed into event_h5’s dtheta_deg
- **liso_c** – The critical luminosity that has to be achieved in order for the function to close a given time bin
- **unit** – A string of the units that will be used for the light curve/spectrum ‘erg/s’ or ‘cts/s’
- **plotting** – Switch to allow the function to show the plots of the light curve/spectral parameters
- **save_plot** – Switch to allow the function to save the produced plot
- **choose_best** – Switch that permits the function to overwrite spectral parameters that do not have error bars to be ignored if set to True. If this is set to False all spectral parameters will be considered/plotted
- **lc_only** – switch that tells the function to only plot the light curve and peak energy of the time resolved spectra and none of the other parameters
- **pol_only** – Switch to allow the function to only plot the light curve and the polarization when set to True
- **dim** – The number of dimensions of the MCRaT simulation that is being analyzed. There is no support for 3D at this time so this switch should be set to 2, which is it’s default value
- **hdf5** – Switch to specify if the MCRaT output files were HDF5 format or not, default is True. Should be set to False when an old version of MCRaT was used which outputted the data in text files
- **save_spex** – Switch that allows the time resolved spectra to be saved to a text file
- **photon_type** – can be set to ‘s’, ‘i’, or left as None in order to select thermal synchrotron photons, injected photons, or all the photons in the simulation for analysis
- **energy_range** – has units of keV, can be left as None to choose photons of all energy ranges for analysis or it can be set to an array with [min energy, max energy] e.g. [1, 10] for 1 to 10 keV (limits inclusive)

Returns returns arrays of the various time resolved parameters and their errors. The order of the variables returned are: light curve, light curve error, alpha,beta,break energy, errors in the spectral parameters (a (times.size,3,2) array where times is the start of the light curve time bins), start times of the time bins, the best fit model in each time bin, polarization, stokes I, stokes Q, stokes U, stokes V, polarization and polarization angle errors (a ((times.size,2)) array), polarization angle, number of photons in each time bin

`read_process_files.lundmanPolarizationGamma(p, theta_j, gamma_0, max_angle)`

Function that calculates Lundman et al.'s (2014b) lorentz factor profile as a function of theta

Parameters

- **p** – The p value for the width of the analytic jet from Lundman et al. (2014b)
- **theta_j** – The opening angle of the analytic jet profile from Lundman et al. (2014b)
- **gamma_0** – The jet core lorentz factor
- **max_angle** – The maximum angle to calculate the lorentz factor of

Returns Returns arrays of the angles and the lorentz factors at those angles

`read_process_files.param_err(x, data, err, best_par, chi_sq, par)`

Function to calculate the 1 sigma errors in the comptonized fit to the spectrum.

Parameters

- **x** – energies of the spectrum in keV
- **data** – The spectrum data points that are being fitted
- **err** – the errors in each data point
- **best_par** – the best fit parameters of the Band function to the MCRaT spectrum
- **chi_sq** – The value of chi squared of the best fit (value of best_par)
- **par** – The Band spectrum parameter of interest that we would like to find the errors of

Returns returns a (1x2) array of the negative error bar and the positive error bar

`read_process_files.param_err_comp(x, data, err, best_par, chi_sq, par)`

Function to calculate the errors in the comptonized fit to the spectrum. Returns 1 sigma error bars.

Parameters

- **x** – energies of the spectrum in keV
- **data** – The spectrum data points that are being fitted
- **err** – the errors in each data point
- **best_par** – the best fit parameters of the COMP function to the MCRaT spectrum
- **chi_sq** – The value of chi squared of the best fit (value of best_par)
- **par** – The COMP spectrum parameter of interest that we would like to find the errors of ('a', 'b', 'c' which corresponds to the definitions of the COMP function)

Returns returns a (1x2) array of the negative error bar and the positive error bar

`read_process_files.param_err_loop(x, data, err, best_par, chi_sq, par, d_par)`

Function to calculate the deviation in a Band spectrum parameter such that there is a change of ~1 in the chi squared value of the fit.

Parameters

- **x** – energies of the spectrum in keV

- **data** – The spectrum data points that are being fitted
- **err** – the errors in each data point
- **best_par** – the best fit parameters of the Band function to the MCRaT spectrum
- **chi_sq** – The value of chi squared of the best fit (value of best_par)
- **par** – The Band spectrum parameter of interest that we would like to find the errors of
- **d_par** – The initial guess in the change in parameter that would lead to a change in chi squared of ~ 1

Returns returns the amount the parameter of interest can change before the fitted chi squared changes by ~ 1

`read_process_files.param_err_loop_comp(x, data, err, best_par, chi_sq, par, d_par)`

Function to calculate the deviation in a COMP spectrum parameter such that there is a change of ~ 1 in the chi squared value of the fit.

Parameters

- **x** – energies of the spectrum in keV
- **data** – The spectrum data points that are being fitted
- **err** – the errors in each data point
- **best_par** – the best fit parameters of the COMP function to the MCRaT spectrum
- **chi_sq** – The value of chi squared of the best fit (value of best_par)
- **par** – The COMP spectrum parameter of interest that we would like to find the errors of
- **d_par** – The initial guess in the change in parameter that would lead to a change in chi squared of ~ 1

Returns returns the amount the parameter of interest can change before the fitted chi squared changes by ~ 1

`read_process_files.plot_golenetskii(sims, t_end, delta_t=1, detectable=False, save_plot=False, h5=True)`

Function to plot the MCRaT results alongside the golenetskii relationship

Parameters

- **sims** – An array containing strings of event file base names that will be plotted
- **t_end** – An array of times that the light curves of the above sim values end
- **delta_t** – The size of the time binds used to get the synthetic GRB light curve/time resolved spectra
- **detectable** – Switch to plot the luminosities below 10^{50} ergs/s or not, set to False by default
- **save_plot** – Switch to save the plot as a PDF or not
- **h5** – Switch to specify if the MCRaT output files were HDF5 format or not, default is True. Should be set to False when an old version of MCRaT was used which outputted the data in text files

Returns

`read_process_files.plotfluidGammaVsTheta` (*derivative=False, comp_lundman=False, save-plot=False*)

This function plots the lorentz factor profile of hydrodynamic outflows, as a function of theta, alongside Lundman et al.'s (2014b) lorentz factor profile

Parameters

- **derivative** – Switch to plot the derivative of the lorentz factor with respect to theta
- **comp_lundman** – Switch to compare the hydro simulation to the lorentz factor profile from Lundman et al. (2014b)
- **saveplot** – Switch to save the plot as a PDF or not

Returns

`read_process_files.polarizationVsAngle` (*event_files, time_start, time_end, dt=1, phi_angle=0, delta_theta=1, unit='erg/s', plotting=True, save_plot=False, dim=2, hdf5=True, compare_to_lund=False, p=4, thetaj=0.01, plot_lumi=False*)

Function to plot how time integrated polarization of the MCraT simulation changes as a function of observer angle. There are option to compare the results to those of Lundman et al. (2014b) and to also overplot the peak luminosity of the light curve at the given angle.

Parameters

- **event_files** – An array of the MCraT event file base names that are numerically ordered from smallest to largest
- **time_start** – An array of the start times of the light curves for the event files specified above
- **time_end** – An array of the end times of the light curves for the event files specified above
- **dt** – If also plotting the peak of the light curve, dt specifies the width of the time bins of the light curves
- **phi_angle** – A placeholder variable for future 3D compatability
- **delta_theta** – delta theta of the observer viewieng angle for accepting photons, also in degrees. Should be the same as what was used in the function call to produce the event files passed to this function
- **unit** – The units of the produced light curve if plotting the light curve peak as well. The default is 'erg/s' but can also pass 'cts/s'
- **plotting** – Switch, set to True by default, to show the plot produced
- **save_plot** – Switch to save the plot as a PDF or not
- **dim** – Sets the number of dimensions of the MCraT simulation analyzed, should be set to 2 since there is no 3d compatability yet.
- **hdf5** – Switch to specify if the MCraT output files were HDF5 format or not, default is True. Should be set to False when an old version of MCraT was used which outputted the data in text files
- **compare_to_lund** – Switch to specify if the user wants to compare their results to those of Lundman et al. (2014b)
- **p** – If compare_to_lund is set to True, the user needs to specify the proper p value for the lorentz factor profile. The default is p=4, also accepts p=2.

- **theta_j** – If compare_to_lund is set to True, the user needs to specify the opening angle of the jet in radians, the default is set to 0.01 rad, also accepts 0.1 rad when p=4.
- **plot_lumi** – A switch to plot the peak luminosity of the light curve associated with the passed event_files

Returns Arrays of the plotted time integrated polarizations and their errors and the polarization angles and their associated 1 sigma errors

`read_process_files.read_flash(fnam, length_scale=1000000000.0, make1D=True)`

Legacy code that was used to read in FLASH data frames in the old python version of the code, still is used in some functions comparing MCRaT information to fluid quantities

Parameters

- **fnam** – the file name fo the FLASH file, including directory it is located in
- **make1D** – switch to set the output to be either 1D, if set to True (the default), or keep the returned information as 2D arrays

Returns returns all of the FLASH file information (x, y, dx, dy, vx, vy, gamma, density, lab density, radius, theta, pressure) at each hydro node

`read_process_files.read_mcrat(file_name)`

Legacy code to read in MCRaT data when it used to output data in text files

Parameters **file_name** – the file name of the MCRaT data file including the directory that it is located

Returns returns all the data in the given file of interest

`read_process_files.read_mcrat_h5(file_name, read_comv=False, read_stokes=False, read_type=False)`

Reads in MCRaT data for current version of MCRaT that outputs data in hdf5 files. Also has support for various MCRaT switches that can be turned on by the user.

Parameters

- **file_name** – the file name of the MCRaT data file including the directory that it is located
- **read_comv** – switch that lets the function know if it should expect/ return comoving 4 momenta data, set to true if this switch is set to ON in mcrat_input.h
- **read_stokes** – switch that lets the function know if it should expect/ return stokes parameters, set to true if this switch is set to ON in mcrat_input.h
- **read_type** – switch that lets the function know if it should expect/ return photon type, set to true if this switch is set to ON in mcrat_input.h

Returns returns the data read in from the MCRaT data frame

`read_process_files.wien(x, t, d)`

Function that produces a Wien spectrum.

Parameters

- **x** – Energy in keV
- **t** – Temp in kelvin
- **d** – Normalization of the spectrum

Returns returns the Wien spectrum parameterized by some temperature at the energies of interest

MCRAT_MOVIE.PY

This routine allows the user to make movies of the photons propagating through the hydro simulation. This routine can also plot the average temperature of the photons vs the average temperature of the matter near the photons

currently this code only works with FLASH hydrodynamic simulations

1st: read FLASH File and MCRaT file

2nd: plot flash temperature and mcrat positions of a bunch of photons

3rd: plot temp vs radius for photons and matter

`mcrat_movie.calc_spectrum(p0, weight)`
produces spectrum in keV

Parameters

- `p0` –
- `weight` –

Returns

`mcrat_movie.follow_lc_data(hydro_sim_dir, hydro_sim_name, last_frame, mcrat_sim_data_dir, maxx, miny, maxy, theta, time, dt, fps, num_photon_plot=10000000.0, read_type=False, energy_range=None)`

Parameters

- `hydro_sim_dir` –
- `hydro_sim_name` –
- `last_frame` –
- `mcrat_sim_data_dir` –
- `maxx` – in real units, not code units
- `miny` –
- `maxy` –
- `theta` – in degrees
- `time` –
- `dt` –
- `fps` –
- `num_photon_plot` –
- `read_type` –

Returns

`mcrat_movie.get_flash_data` (*file_num, f, max_x, min_y, max_y, flash_l_scale*)

This function collects all the data from a given FLASH file and prepares it to be plotted with imshow.

Parameters

- **file_num** – The frame number
- **f** – The path to the file as well as the FLASH file name preceeding the file number
- **max_x** – The maximum x that should be plotted (the input should be in physical units, not in code units)
- **min_y** – The minimum y of the portion of the hydro frame that will be plotted (the input should be in code units)
- **max_y** – The maximum y of the portion of the hydro frame that will be plotted (the input should be in code units)

Returns returns a 2D array of the density for plotting with imshow

`mcrat_movie.get_indexes_data` (*mcrat_f, num, ph_num, angle, t, dt, fps, read_comv=False, read_stokes=False, read_type=False, dtheta_deg=1, energy_range=None*)

This function gets the set of indexes of the photons that will be plotted in the imshow window as well as the indexes of all the photons in order to calculate the spectrum

Parameters

- **mcrat_f** –
- **num** –
- **ph_num** –
- **angle** –
- **t** –
- **dt** –
- **fps** –
- **read_comv** –
- **read_stokes** –
- **dtheta_deg** –

Returns

`mcrat_movie.get_plot_data` (*frame_number, flash_file, mcrat_file, t_ph=None, t_f=None, plot_temp_curves=False, read_comv=False, read_stokes=False, read_type=False, dtheta_deg=1*)

Parameters

- **frame_number** –
- **flash_file** –
- **mcrat_file** –
- **t_ph** –
- **t_f** –
- **plot_temp_curves** –

- `read_comv` –
- `read_stokes` –
- `read_type` –
- `dtheta_deg` –

Returns

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

m

mclib, [1](#)
mcrat_movie, [19](#)

r

read_process_files, [7](#)

A

amati() (in module read_process_files), 7

B

Band() (in module read_process_files), 7
band_hist_data() (in module read_process_files), 7
band_hist_plot() (in module read_process_files), 8
blackbody() (in module read_process_files), 8

C

calc_spectrum() (in module mcrat_movie), 19
cfits() (in module read_process_files), 8
comp() (in module read_process_files), 9

E

event3D() (in module mclib), 1
event4() (in module mclib), 1
event_h5() (in module mclib), 2

F

fluidGammaVsTheta() (in module read_process_files), 9
follow_lc_data() (in module mcrat_movie), 19

G

get_amati_rel() (in module read_process_files), 10
get_FERMI_best_data() (in module read_process_files), 9
get_flash_data() (in module mcrat_movie), 20
get_indexes_data() (in module mcrat_movie), 20
get_lundman_pol_data() (in module read_process_files), 10
get_plot_data() (in module mcrat_movie), 20
get_yonetoku_rel() (in module read_process_files), 10
golenetskii() (in module read_process_files), 10
Goodman() (in module read_process_files), 7

L

lcur() (in module mclib), 2
lcur_old() (in module mclib), 3

lcur_param() (in module read_process_files), 11
lcur_param_plot() (in module read_process_files), 12
lcur_param_var_t() (in module read_process_files), 12
lcur_var_t() (in module mclib), 3
Lorentz_Boost() (in module mclib), 1
lorentzBoostVectorized() (in module mclib), 4
lundmanPolarizationGamma() (in module read_process_files), 14

M

mclib(module), 1
mcrat_movie(module), 19

P

param_err() (in module read_process_files), 14
param_err_comp() (in module read_process_files), 14
param_err_loop() (in module read_process_files), 14
param_err_loop_comp() (in module read_process_files), 15
plot_golenetskii() (in module read_process_files), 15
plotfluidGammaVsTheta() (in module read_process_files), 15
polarizationVsAngle() (in module read_process_files), 16

R

read_event_file() (in module mclib), 4
read_flash() (in module read_process_files), 17
read_mcrat() (in module read_process_files), 17
read_mcrat_h5() (in module read_process_files), 17
read_process_files(module), 7
readanddecimate() (in module mclib), 5

S

single_cs() (in module mclib), 5
single_electron() (in module mclib), 5
spex() (in module mclib), 5

W

`wien()` (*in module `read_process_files`*), [17](#)

Z

`zero_norm()` (*in module `mclib`*), [6](#)