RefleX User Manual

Version 2.1

Stéphane Paltani & Claudio Ricci
Astronomy & Astrophysics, 2017, 607, A31

May 22, 2019

Abstract

RefleX is a ray-tracing code designed for X-ray photons (energies from about 100 eV up to about 1 MeV). Photons can be generated with different source geometries and different emission spectrum. The photons encounter different user-specified objects. The following processes are implemented: photo-ionization, Compton scattering on free electrons, Compton scattering on bound electrons, Rayleigh scattering, K-shell and L-shell fluorescence and inverse Compton scattering. RefleX can output its results in a very versatile way, either in text or in binary, either photon by photon or binned in user-defined spectra.

1 What's new in RefleX 2.1?

This is a minor release, with no significant effect on the results. Users who performs simulations with many objects need however to upgrade to RefleX 2.1 for computation efficiency (see last bullet point below).

Physics

- H_2 fraction is treated more consistently.

Control

- RefleX can now output the computed cross-sections.
- Fixed a bug which forced recomputing cross-sections for all objects, even if the properties are unchanged.

2 Invocation

RefleX is called the following way:

reflex [parameter_file]*

An arbitrary (larger than or equal to 0) number of parameter files can be passed on the command line; they will be processed sequentially. If the `parameter_file` parameter is left out, the file `reflex.par` is used.

Please report any issue about RefleX to `Stephane.Paltani@unige.ch`

3 Basic principles

3.1 Ingredients

RefleX needs the following ingredients to be described in the parameter file(s) (See Sect. **??**):

- One source geometry
- One source spectral shape
- One domain in which the simulation takes place, called "World"
- Any number of objects in which photons are propagated
- The composition and density of these objects
- The physical processes that need to be simulated (although the default can often be used)
- The duration of the simulation, either a number of photons or a duration
- Optionally (but this is usually needed) some directives on how to output the results
- Optionally, some control parameters that affect the working of RefleX

3.2 Physics

The physical processes implemented in Reflex are described below. By default, they are all applied, but they can be turned off or modified by the user.

- Bound/free electrons
  Two physical configurations can be selected by setting the 'temperature' of the medium: If the temperature is cold, all electrons are bound to atoms; this is the atomic configuration. If the temperature is 'warm', Hydrogen and Helium atoms are supposed to be fully ionized. Electrons from other elements are ignored; this is the free-electron configuration. Each object can use a different temperature. In the cold case, some user-specified fraction of Hydrogen is in the form of molecular Hydrogen, while in the warm case, all H_2 are destroyed.

- Photoionization
  Photoionization is implemented using the cross-sections of Verner & Yakovlev, 1995,A&AS 109, 125 and Verner et al., 1996, ApJ 465, 487. It is limited to $Z \leq 30$, i.e. Zinc.

- Compton scattering
  Compton scattering is implemented for free electrons using the full Klein-Nishina differential cross-sections, as well as for bound electrons using the tabulated cross-sections from Hubbell et al., 1975, J. Phys. Chem. Ref. Data 4, 471 (which converge to Klein-Nishina cross-sections at

high energies). For education purpose, Klein-Nishina cross-sections can be turned off (and use Thomson cross-section at all energies); Compton scattering can also be replaced by Thomson (elastic) scattering.

- Rayleigh scattering
  Rayleigh scattering is applied only in the atomic configuration (cold temperature), and uses the tabulated cross-sections from Hubbell et al., 1975, J. Phys. Chem. Ref. Data 4, 471

- Fluorescence emission
  Both K- and L-shell fluorescence is included. Energy levels are taken from Bearden, 1967, Rev. Mod. Phys., 39, 78 when available and from the EPDL89 library (Cullen et al., 1990, UCRL-ID-103424. Transition probabilities are from EPDL89. Coster-Kronig process is not implemented.

- Polarization
  Optionally, polarization can be taken into account in Compton scattering. Note however that this functionality has not been validated, and is currently only indicative.

3.3 Objects

In addition to their shape and size, objects are given a chemical composition, which can be composed element by element (some compositions are predefined), and a density. Each object can have different compositions, different densities and physical conditions. Each object can also be named in order to trace the photons.

The user needs to set first the parameters determining the content of the objects: composition, density and physical conditions. When an object is created, it inherits the pameters that have been previously set, as well as the whole information about the cross-sections of all physical processes, as selected by the user. All subsequently created objects inherit the same properties, until a change is detected.

The cross-sections are recomputed only if the composition, density, or the physical conditions have changed. Thus, when creating a large number of objects (e.g., a clumpy torus), the time to compute all cross-sections and the amount of memory needed to store the information can become very large if each clump is set with different properties. In such situation, the user can decrease the resolution of the grid.

3.4 Units, axes and assumptions

- The lengths can be expressed either in meter, centimeter, light-year, light-second, or parsec.
- Densities can be expressed either in meter^{-3}, centimeter^{-3}, light-year^{-3} or parsec^{-3}.
- Time is in seconds.
- Energies and temperatures are in eV.
- Angles are expressed in degrees
- The X axis of Cartesian coordinates (1,0,0) is the polar vector $(90, 0)$. The Z axis of Cartesian coordinates (0,0,1) is the polar vector $(0, x)$ (x is arbitrary).

- Boolean parameters are either "ON" or "OFF".
- Only (and all) chemical elements from Z=1 to Z=30 are considered.

3.5 Output

When run, RefleX outputs its log through STDERR; it contains all the information relative to the processing which is contained in the parameter file. This is used to provide metadata and to ensure that the parameter file has been interpreted correctly.

RefleX has four different ways to output the result of the simulations; all of them can be used at the same time:

- Text : Each event affecting the simulated photon can be output. The user can select which of these events is displayed. The text output is normally sent to STDOUT, but it can be output to a file instead. This is the output type that contains the most information; it is generally used for debugging or educational purpose. The following event types can be selected:
  - Photon hits World
  - Photon is created
  - Photon is destroyed
  - Photon enters an object
  - Photon exits an object
  - Photon makes a fluorescence
  - Photon makes a Compton scattering
  - Photon makes a Rayleigh scattering
- Binary : The properties of photons hitting World can be output in binary format. The user can select which of the fields are output. The binary output must be written to a file. Any selection of the photon fields can be specified. These fields are:
  - Number of Compton scatterings
  - Number of Rayleigh scatterings
  - Number of fluorescence emissions
  - Numbers of hit objects
  - Time when the photon hits World
  - Photon energy
  - Photon direction (X, Y, Z) coordinates, $||(X, Y, Z)|| = 1$
  - Photon location (X, Y, Z) coordinates, $||(X, Y, Z)||$ =World radius
  - Photon polarization (X, Y, Z) coordinates, $||(X, Y, Z)|| = 1$, perpendicular to direction
- Spectral output: Photons hitting World can be selected using arbitrary criteria and binned into spectra with arbitrary energy bins (linear or logarithmic). An arbitrary number of spectra can be created in each run of RefleX. Spectral outputs must be written to a file. Photons can be selected according to the following properties:

  - Number of interactions
  - Number of Compton scatterings
  - Number of Rayleigh scatterings
  - Number of Compton or Rayleigh scatterings
  - Number of fluorescence emissions
  - Numbers of hit objects
  - Time when the photon hits World
  - Photon energy
  - Photon direction (X, Y, Z) coordinates, each coordinate individually
  - Photon location (X, Y, Z) coordinates, each coordinate individually
  - Photon polarization coordinates (X, Y, Z), each coordinate individually

  When several constraints are provided, they are combined with AND.

- FITS image: The user can place a "detector", which, when hit by a photon, records the number of photons. An arbitrary number of images can be created in each run of RefleX. Image outputs must be written to a file. Photons can be selected according to the following properties:

  - Number of interactions
  - Number of Compton scatterings
  - Number of Rayleigh scatterings
  - Number of Compton or Rayleigh scatterings
  - Number of fluorescence emissions
  - Numbers of hit objects
  - Time when the photon hits World
  - Photon energy
  - Photon direction (X, Y, Z) coordinates, each coordinate individually
  - Photon location (X, Y, Z) coordinates, each coordinate individually
  - Photon polarization coordinates (X, Y, Z), each coordinate individually

  When several constraints are provided, they are combined with AND.

3.6 Fluxes

The user can set the intrinsic source luminosity in a given energy range. This implies that the redshift of the source is provided. Cosmology can be set by providing H_0, Ω_M and Ω_Λ. In spectral output mode, the user can optionally produce photon flux spectra in $\mathrm{cm}^{-2}\ \mathrm{s}^{-1}\ \mathrm{keV}^{-1}$ or energy flux spectra in $\mathrm{keV}\ \mathrm{cm}^{-2}\ \mathrm{s}^{-1}\ \mathrm{keV}^{-1}$. In text and binary output mode, the time needed for a real source to generate the requested number of photons and the flux associated to each event are provided in the log output of RefleX (through STDERR). This flux is in $\mathrm{cm}^{-2}\ \mathrm{s}^{-1}$.

4 Parameter file

The parameter file is a series of ASCII directives. The commands are grouped into several categories: general setup commands, object setup commands, physics commands, output commands. Commands marked with a star (*) can be repeated several times. Commands have an effect on all subsequent commands (until changed).

4.1 General setup commands

- **RANDOM** *INT*

  When specified, "RANDOM" sets the seed of the random number generator to the parameter. If not specified, an unnown seed is used, so the results are not reproducible

- **NPHOTS** *INT*

  Sets the number of photons in the simulation. If it is 0, the number of photons is infinite and the simulation must be stopped by a time limit or by sending SIGINT or SIGTERM.

- **TIME** *INT*

  Sets the duration of the simulation in seconds. If it is 0 (the default value), the duration is infinite the simulation must be stopped by a limit in the number of photons or by sending SIGINT or SIGTERM.

- ***LENGTH** *STRING*

  Sets the unit of all length (and derived, such as densities) directives in the parameter file. The value can be:

  - **Meter** (default)
  - **Centimeter**
  - **Lightyear**
  - **Lightsecond**
  - **Parsec**

- **THREAD** *INT*=N

  Runs RefleX using N threads. There is no checks that the processor can handle these threads, so N should be adapted to the processor by the user. Note that in spite of careful coding of the multithreaded parts of the code, RefleX still experiences some defects when run in multithreaded mode. Thus this should be used for exploration only.

- **MAXINT** *INT*

  Defines the maximum number of interactions that a photon can undergo until it is declared lost. The default is 10000. Normally, MAXINT is never reached. However, in some specific geometry, a photon might get "stuck" in an object because of numerical precision issues; MAXINT is able to kill such photons, in order not to block the simulations. This should be very rare, so it is harmless. There should be no need to change the value of MAXINT. The number of killed photons is reported by RefleX.

- **MARGIN** *FLOAT*

  In order to further prevent photons to get "stuck" in an object, MARGIN defines the minimum distance that a photon has to cross at each iteration. It is expressed as a fraction of the size of the object, and the default is 10^{-12}. There should be no need to change the value of MARGIN.

- **STEP** *FLOAT*

  Defines the energy resolution of the cross-section computations below 10 keV. It is expressed in eV and the default value is 0.1 eV. Above 10 keV, since there is no atomic transition anymore, the step is increased by a factor 100. Computation time does not depend dramatically on the value of STEP, except when creating the grid (it takes however only a few seconds per grid). However, in simulations with many objects with different compositions or physical conditions, the grids might occupy a lot of memory. With the default resolution of 0.1 keV, each grid takes about 50 MB; it would be only 5 MB for a 1 eV resolution.

4.2 Object setup commands

- ***OBJECT** *STRING* ...

  Add an object. The parameters can be any of the following:

  - **WORLD** *FLOAT*=R

    WORLD is a special object that defines the domain where simulations are run. It is a sphere centered on (0,0,0) and of radius R. WORLD is mandatory.

  - **SPHERE** *STRING*=N *FLOAT*=X *FLOAT*=Y *FLOAT*=Z *FLOAT*=R

    The object is a sphere of center (X, Y, Z) and radius R. It is identified by its name N.

  - **DISC** *STRING*=N *FLOAT*=X *FLOAT*=Y *FLOAT*=Z *FLOAT*=R *FLOAT*=H

    The object is a disc of center (X, Y, Z), radius R and total height H. The plane of the disc is parallel to the ((1,0,0),(0.1,0)) plane (XY plane). It is identified by its name N.

  - **ANNULUS** *STRING*=N *FLOAT*=X *FLOAT*=Y *FLOAT*=Z *FLOAT*=R *FLOAT*=R_{in} *FLOAT*=H

    The object is an annulus of center (X, Y, Z), radius R and total height H. The inner radius of the annulus is R_{in}. The plane of the disc is parallel to the ((1,0,0),(0.1,0)) plane (XY plane). It is identified by its name N.

  - **TORUS** *STRING*=N *FLOAT*=X *FLOAT*=Y *FLOAT*=Z *FLOAT*=R *FLOAT*=R_{in}

    The object is a torus of center $(0, 0, 0)$, external radius R and internal radius R_{in}. The plane of the torus must be parallel to the ((1,0,0),(0.1,0)) plane (XY plane). It is identified by its name N.

  - **CONE** *STRING*=N *FLOAT*=X *FLOAT*=Y *FLOAT*=Z *FLOAT*=B *FLOAT*=T *FLOAT*=$\mathrm{R_B}$ *FLOAT*=$\mathrm{R_T}$

    The object is a cone of center $(0, 0, 0)$, which starts at height B up to height T. The bottom radius is $\mathrm{R_B}$, while the top radius is $\mathrm{R_T}$. The cone is around the Z axis (0,0,1). It is identified by its name N.

  - **HCONE** *STRING*=N *FLOAT*=X *FLOAT*=Y *FLOAT*=Z *FLOAT*=B *FLOAT*=T *FLOAT*=$\mathrm{R_{B}}_{out}$ *FLOAT*=$\mathrm{R_{T}}_{out}$ *FLOAT*=$\mathrm{R_{B}}_{in}$ *FLOAT*=$\mathrm{R_{T}}_{in}$

The object is a hollow cone of center $(0, 0, 0)$, which starts at height B up to height T. The outer bottom radius is R_{Bout}, while the outer top radius is R_{Tout}. The inner bottom radius is R_{Bin}, while the inner top radius is R_{Tin}. The hollow cone is around the Z axis (0,0,1). It is identified by its name N.

- *__DENSITY__ *FLOAT*

  Sets the density of Hydrogen atoms in units of 1/LENGTH^3

- *__MATTER__ *STRING*

  Loads a predefined composition. *STRING* can currently be:

  - **lodd**: Lodders, ApJ 591, 1220 (2003)
  - **angr**: Anders & Grevesse, Geochemica and Cosmochimica Acta (1989).

- *__METALLICITY__ *FLOAT*

  Sets the metallicity to *FLOAT*, i.e., multiply the fraction of elements from Z=3 to Z=30 by *FLOAT*.

- *__ELEMENT__ *INT FLOAT*

  Sets the abundance of element Z=*INT* to *FLOAT*, i.e., multiply the fraction of elemen Z=*INT* by *FLOAT*.

- *__TEMPERATURE__ *FLOAT*=V

  Sets the temperature in the subsequent objects to V (in eV), if V> 1. In this case only inverse Compton scattering happens in the object(s). If V= 0, the medium is cold, and all atoms are considered neutral, so Compton and Rayleigh scattering happens on bound electrons only. If V= 1, the medium is warm and Hydrogen and Helium are considered to be fully ionized, consistent with the assumption in pexrav, pexmon MYTorus, etc.

- *__H2FRACTION__ *FLOAT*=F

  Sets the fraction of Hydrogen in molecular form. One must have 0 <F< 1. If the medium is warm, F is set to 0. Attempts to change it results in an error.

4.3 Object setup commands

- __EMSPEC__ *STRING ...*

  Sets the emission spectrum of the source. The parameters can be:

  - **MONO** *FLOAT*=E: Generates mono-energetic photons at energy E eV
  - **GAUSS** *FLOAT*=E *FLOAT*=S: Generates photons following a Gaussian distribution cenetered on E eV, with an rms S eV
  - **PWRLAW** *FLOAT*=G: Generates photons following a power-law with index G
  - **CUTOFF** *FLOAT*=G *FLOAT*=C: Generates photons following a cut-off power-law with index G and cut-off energy C eV
  - **BLACKBODY** *FLOAT*=T: Generates photons following a blackbody distribution with temperature T eV

- **WIEN** *FLOAT*=T: Generates photons following a Wien distribution with temperature T eV

- **EMGEOM** *STRING* ... Sets the geometry of the source. The parameter can be:
  - **POINT** *FLOAT*=X *FLOAT*=Y *FLOAT*=Z *FLOAT*=ϑ *FLOAT*=φ *FLOAT*=α: The source is a point source centered on (X,Y,Z), and photons are emitted in a cone around (ϑ, φ) deg, with an opening angle of α deg. If $\alpha = 0$, all photons have the direction (ϑ, φ). The vector (0,0,1) correspond to the direction $(0, x)$.
  - **SPHERE** *FLOAT*=X *FLOAT*=Y *FLOAT*=Z *FLOAT*=R *FLOAT*=ϑ *FLOAT*=φ *FLOAT*=α: The source is a sphere centered on (X,Y,Z) with a radius R, and photons are emitted in all directions on a cone around (ϑ, φ) deg, with an opening angle of α deg. If $\alpha = 0$, all photons are emitted from the point $R(\vartheta, \varphi)$. The vector (0,0,1) correspond to the direction $(0, x)$.
  - **DISC** *FLOAT*=X *FLOAT*=Y *FLOAT*=Z *FLOAT*=R *FLOAT*=D: The source is a disc in the (X, Y) plane centered on (X,Y,Z) and radius R, and photons are emitted above the disc if D>0 and below if D<0.

- **ECUT** *FLOAT*

  Sets the photon termination energy in eV, i.e., the photon is destroyed if its energy falls below this value

- **EGEN** *FLOAT*=E_1 *FLOAT*=E_2

  Generates photons in the range (E_1,E_2) eV only

- **PHYSICS** *BOOL*=B

  Turns on (B="ON") or off (B="OFF") the physical process P. There can be several 'PHYSICS ...' commands; however they have an effect over the entire simulation (i.e., physical processes cannot be turned on and off for each object individually). P can be:
  - **COMPTON** : Turns on Compton scattering. This is ON by default.
  - **RAYLEIGH** : Turns on Rayleigh scattering. This is ON by default.
  - **PHOTO** : Calculates photo-ionization cross-sections. This is ON by default.
  - **KN** : Uses the Klein-Nishina cross-sections for free electrons. This is ON by default. If OFF, uses the Thompson cross-section at all energies.
  - **THOMSON** : Uses Thomson cross-sections at all energies and treat all scatterings as elastic. This is OFF by default.
  - **FLUOR** : Uses the fluorescence yields after photo-ionization. This is ON by default.
  - **FIRON** : Discards fluorescence from elements other than Iron and Nickel. This is OFF by default.
  - **POLAR** : Turns on polarization effects in the ray tracing. This is OFF by default.

- **ABUNDMIN** *FLOAT*

  Exclude all elements whose abundance is lower than *FLOAT* in all cross-section calculations

- **LUMINOSITY** *FLOAT*=E_{min} *FLOAT*=E_{max} *FLOAT*=Log L

  Sets the source intrinsic luminosity between E_{min} eV and E_{max} eV to Log L, with L expressed in erg s^{-1}.

- **REDSHIFT** *FLOAT*

  Sets the source's redshift to *FLOAT*

- **COSMOLOGY** *FLOAT*=H_0 *FLOAT*=Ω_M *FLOAT*=Ω_Λ Sets the cosmological parameters to H_0 km s^{-1} Mpc^{-1}, Ω_M and Ω_Λ

4.4 Output commands

- **VERBOSE** *INT*

  Define the verbosity of the Text output. The parameter defines which event is printed. It is bit-encoded:

  - Bit 0 : Photon hits World
  - Bit 1 : Photon is created
  - Bit 2 : Photon is destroyed
  - Bit 3 : Photon enters an object
  - Bit 4 : Photon exits an object
  - Bit 5 : Photon makes a fluorescence
  - Bit 6 : Photon makes a Compton scattering
  - Bit 7 : Photon makes a Rayleigh scattering
  - Bit 8 : Photon makes an inverse Compton scattering

  For example, "VERBOSE 11" will print information when a photon is created, when a photon enters an object or when the photon hits World.

- **PERCENT** *BOOL*

  Print out the percentage of processed photons along the way

- **FRESET** *BOOL*

  If "ON", resets the number of scatterings (Compton and Rayleigh) to 0 after a fluorescence event. This allows to determine the scattering order of photons generated with fluorescence.

- **OUTNAME** *STRING*

  Sets the name of the Text output file to *STRING*. If not set, all out put goes to STDOUT.

- **OUTDATA** *STRING*

  Sets the name of the Binary output file to *STRING*. If not set, no Binary output is produced.

- **FIELDS** *STRING*

  Defines which photon fields are written in the Binary output file. It must be a sequence of 15 "0" and "1", "1" indicating that the field is output. The fields are :

- Digit 1: Number of Compton scatterings
- Digit 2: Number of Rayleigh scatterings
- Digit 3: Number of fluorescence emissions
- Digit 4: Numbers of hit objects
- Digit 5: Time when the photon hits World
- Digit 6: Photon energy
- Digit 7: Photon direction X coordinate
- Digit 8: Photon direction Y coordinate
- Digit 9: Photon direction Z coordinate
- Digit 10: Photon location X coordinate
- Digit 11: Photon location Y coordinate
- Digit 12: Photon location Z coordinate
- Digit 13: Photon polarization X coordinate
- Digit 14: Photon polarization Y coordinate
- Digit 15: Photon polarization Z coordinate

- *__DUMP__ *STRING* *INT*=INT_MIN *INT*=INT_MAX

  Dumps the full current cross-sections for the specified processes. Cross-sections from processes from INT_MIN to INT_MAX are summed, with the process number being:

  - 0: H_2 photoionization cross-section
  - 1–30: photoionization cross-section from element Z
  - 31: Rayleigh cross-section
  - 32: Compton cross-section

- *__SPECTRUM BINNING__ *FLOAT*=E_{min} *FLOAT*=E_{max} *FLOAT*=Step [LOG]

  Defines a new binning for all subsequent spectra from E_{min} eV to E_{max} eV with constant step Step eV. If "LOG" is specified, E_{min} and E_{max} are expressed in (decimal) logarithm and Step is a multiplicative constant.

  *__SPECTRUM NEW__ *STRING*

  Create a new spectrum with filename STRING.

  *__SPECTRUM MODE__ *STRING*

  Set the format of the current spectrum according to STRING, which can be:

  - **COUNTS**: The spectrum consists in integer photon counts (no normalization)
  - **PHOTON**: The spectrum is expressed in cm^{-2} s^{-1} keV^{-1}
  - **FLUX**: The spectrum is expressed in keV cm^{-2} s^{-1} keV^{-1}

Note that the flux spectra (either PHOTON or COUNTS) assume that the spectrum is collected over 4π. If this is not the case, e.g., because the user has selected specific photon directions, it is up to the user to correct the flux for this factor.

***SPECTRUM** *STRING*=Par *STRING*=Cond *FLOAT*=Val

Adds a new condition for the current spectrum. Par can be:

- **COMPTON**: Number of Compton scatterings
- **FLUOR**: Number of fluorescence emissions
- **RAYLEIGH**: Number of Rayleigh scatterings
- **SCATTER**: Total number of scatterings (Compton+Rayleigh)
- **INT**: Total number of "interactions" (Compton+Rayleigh+Fluorescence)
- **OBJECT**: Number of objects hit
- **TIME**: Time when reaching World
- **ENERGY**: Energy when reaching World
- **DIR_X**: X-axis direction when reaching World
- **DIR_Y**: Y-axis direction when reaching World
- **DIR_Z**: Z-axis direction when reaching World
- **LOC_X**: X-axis position when reaching World
- **LOC_Y**: Y-axis position when reaching World
- **LOC_Z**: Z-axis position when reaching World
- **POL_X**: X-axis polarization when reaching World
- **POL_Y**: Y-axis polarization when reaching World
- **POL_Z**: Z-axis polarization when reaching World

The condition Cond can be any of <, <=, ==, ! =, >=, >. The parameter Par is compared to the value Val using the condition Cond.

- ***IMAGE NEW** *STRING FLOAT*=X **AXIS**/*FLOAT*=Y *FLOAT*=Z *FLOAT*=A *FLOAT*=F *INT*=N

  Create a new FITS image with filename STRING. The detector is placed at (X, Y, Z) and is oriented towards the center of the simulation. All photons passing at a distance smaller than A are recorded. Hence A is the aperture of the "camera"; a larger aperture will increase luminosity, but at the expense of image resolution. The image is a square of NxN pixels, and has a field-of-view of F degrees.

  If the fourth field is AXIS, the detector is placed at coordinates (X,0,Z), and then rotated around the Z axis. This is equivalent to placing an infinite mount of repeated detectors (see IMAGE REPEAT) at all possible coordinates (x,$\sqrt{X^2 - x^2}$,Z), without the need of adding numerous detectors. Using this feature makes sense only if the system is axisymmetric.

  ***IMAGE REPEAT** *FLOAT*=X *FLOAT*=Y *FLOAT*=Z

Add a detector at (X, Y, Z), identical to the one that has been previously defined. Any number of IMAGE REPEAT can be added. This functionality is used to increase the signal in case where the system has a symmetry.

***IMAGE** *STRING*=Par *STRING*=Cond *FLOAT*=Val

Adds a new condition for the current image. Par can be:

- **COMPTON**: Number of Compton scatterings
- **FLUOR**: Number of fluorescence emissions
- **RAYLEIGH**: Number of Rayleigh scatterings
- **SCATTER**: Total number of scatterings (Compton+Rayleigh)
- **INT**: Total number of "interactions" (Compton+Rayleigh+Fluorescence)
- **OBJECT**: Number of objects hit
- **TIME**: Time when reaching World
- **ENERGY**: Energy when reaching World
- **DIR_X**: X-axis direction when reaching World
- **DIR_Y**: Y-axis direction when reaching World
- **DIR_Z**: Z-axis direction when reaching World
- **LOC_X**: X-axis position when reaching World
- **LOC_Y**: Y-axis position when reaching World
- **LOC_Z**: Z-axis position when reaching World
- **POL_X**: X-axis polarization when reaching World
- **POL_Y**: Y-axis polarization when reaching World
- **POL_Z**: Z-axis polarization when reaching World

The condition Cond can be any of <, <=, ==, ! =, >=, >. The parameter Par is compared to the value Val using the condition Cond.

A Changelog

A.1 RefleX 2.0

Optimization

- Major speed improvements thanks to precomputation of all cross-sections, with user-defined accuracy, and a more clever selection of the interactions
- Many bug fixes (not affecting the physics)

Physics

- A Wien input spectrum has been added.
- Setting H_2 fraction is forbidden for warm media.

Geometry

- All objects can now be placed at any position.
- A cone and a hollow-cone geometry have been added.

Images

- Many bug fixes in the creation of images.
- Images of axisymmetric geometries can now make use of this symmetry for increased efficiency.

Units

- Length unit light-seconds has been added.

Control

- RefleX can now use multi-threading, with some caveat.
- RefleX can be stopped after a user-specified number of seconds.
- RefleX can be interrupted by SIGINT or SIGTERM while preserving the computations already achieved.
- Control of stuck photons, which occur sometimes because of rounding errors for complex geometries.

A.2 RefleX 1.0

First released version