

# CoREAS 1.2 User's Manual

Tim Huege\*

June 2, 2017

## 1 What's new in CoREAS V1.2?

The physics of CoREAS is unchanged between version 1.1 and version 1.2. However, a bug was fixed that was present in all previous CoREAS releases. It affected simulations with zenith angles beyond 75 degrees zenith angle, for which a step-wise numerical integration of the refractive index along the line of sight should have been carried out – but was not. Up to 80 degrees zenith angle, the deviations caused by this bug were minor. Simulations with zenith angles beyond 80 degrees zenith angle were significantly affected, though, and should be redone with CoREAS V1.2 or later. Please note that the needed computation time increases significantly for inclined showers due to the stepwise integration. This might be improved in a future version. If job-times get too long to be handled, please have a look at the quick-start guide for the MPI-parallelized version of CoREAS.

The performance of CoREAS V1.2 has been significantly optimized with respect to earlier versions. This has been achieved by tabulating the atmosphere rather than calling CORSIKA-internal functions that rely on calculations of exponential functions. Run-times decrease by 20-30% with this optimization. I would like to thank Tobias Winchen for carrying out the profiling and suggesting the tabulation.

The MPI-parallelized version of CoREAS, originally implemented with the help of Gevorg Poghosyan, has been significantly improved by Elizaveta Dorofeeva. This includes better memory management ensuring better compatibility with a wide range of computing cluster configurations, and a more efficient collection of the results calculated on individual cores, significantly decreasing the time needed for writeout of results. A quick-start guide for running CoREAS with MPI-parallelization is now included in section 9.

A number of technical issues appearing in special computing environments or with very new gcc compilers have also been fixed.

## 2 What was new in CoREAS V1.1?

The physics of CoREAS is unchanged between version 1.0 and version 1.1. All results remain directly comparable. There are, however, two new features:

---

\*email: tim.huege@kit.edu

## 2.1 Parallelized simulation using MPI

CORSIKA can be run in parallel mode on the basis of MPI — please check the various user’s manuals for more information. In parallelized mode, sub-showers of particles are simulated in a deterministic and reproducible way in individual sub-jobs. CoREAS has now been extended to support this parallel mode of simulation.

This is relevant in particular for long-running simulations such as those with very many antenna positions and/or very conservative thinning parameters, where wall-times and other considerations make simulations impractical. Using the parallelized MPI simulation, such jobs can be completed efficiently and quickly within time-scales that can be handled easily. Another application is given by simulations where a quick turnaround is needed, for example if an interactive scheme of modifying simulation parameters is used.

## 2.2 Slicing of simulations

A simple interface for “slicing” simulations has been introduced in CoREAS. This means that antennas can be configured to only accept radio emission contributions from particles in a certain range of Lorentz factors (keyword *gamma*), slant depth (keyword *slantdepth*), geometrical distance along the shower axis from the shower core (keyword *distance*), or vertical height above the observation level (keyword *height*). The syntax to activate this is to append the keyword together with the lower accepted limit and upper accepted limit in the antenna list file. To accept only particles with a Lorentz factor  $\geq 2.0$  and  $< 10.0$ , the entry would be:

```
AntennaPosition = 10000      0  140000  antennaname gamma 2.0 10.0
```

Combinations of sliced antennas and unsliced antennas can be made without any limitations within the same simulation.

## 3 For former REAS3.x users

If you have used REAS3.x before, here is an executive summary of the most important changes between REAS3.x and CoREAS.

- CoREAS uses the same “endpoint formalism” as implemented in REAS3.x. However, instead of storing the air shower in histograms and recreating it from these, individual particle tracks are followed directly during the CORSIKA calculation. This is much faster and much more precise.
- CoREAS is now integrated directly in CORSIKA. You simply compile CORSIKA with the CoREAS option switched on. No environment variables need to be set up, and you do not need ROOT. After a CORSIKA run with the CoREAS option active, the simulated electric field traces will be available directly.

- Input for CoREAS consists of a .list and a .reas file, just as was the case for REAS3.x. The file names for these two files have to correspond to SIMxxxxxx.list and SIMxxxxxx.reas, where xxxxxx is the run-number. The options to configure the CoREAS simulation are a small subset of those that were available for REAS3.x. The output will be written to the directory SIMxxxxxx.coreas. The output format is identical to that of REAS3.x.

## 4 Introduction

CoREAS [1] is a C++ code for the simulation of *CORSIKA*-based *Radio Emission* from Air Showers. It is based on the “endpoint formalism” previously implemented in REAS3.x [9]. In contrast to REAS3.x, the endpoint formalism is now applied “on-the-fly” for each individual particle track followed in the *CORSIKA* air shower simulation. The endpoint formalism provides a universal way of calculating the electromagnetic radiation associated with arbitrarily moving particles and is discussed in more detail in [7]. Due to the universality of the approach, CoREAS therefore automatically incorporates all of the emission associated with the particle motion in air showers. The effect of the atmospheric refractive index is also taken into account in the simulation, leading to “Cherenkov-like” time-compression and amplification effects.

In practice, simulating radio emission from extensive air showers with CoREAS works by running a specially compiled *CORSIKA* binary using the simulation parameters of interest. At the end of the *CORSIKA* run, the electric field time-traces will be written to disk.

The terms and conditions for the usage of CoREAS are detailed in section 11. In any case, please contact the authors if you find bugs or have programmed routines that would be useful to include in the CoREAS distribution.

## 5 Installation

Compile *CORSIKA* as usual by starting

```
cd /home/user/corsika7xxx
./coconut
```

Select the options you would like to use, and make sure to switch on the option

```
c - CoREAS Radio Simulations
```

Finish the selection and let the compilation run through. That’s it. No need to set up any environment variables, no need of a working installation of ROOT.

## 6 Running an example simulation

After finishing the compilation of *CORSIKA*, a binary will be created in the subdirectory *run*, ready for you to use. The usage of *CORSIKA* stays the same as in the standard case, except that you need to supply two additional files to configure the CoREAS part. Please refer to the *CORSIKA* user’s manual for further information on running *CORSIKA* in general. When you run a simulation with CoREAS enabled, there will be a message in the *CORSIKA* log stating so and at the end of the simulation run, *CORSIKA* will create an additional directory *SIMxxxxxx\_coreas*. This directory contains the electric field traces for the radio simulation.

An example has been prepared so that you can immediately run a simulation. After the *CORSIKA* binary has been compiled, do the following:

```
cd run
./corsika7xxx <RUN000001.inp >RUN000001.log
```

where you need to replace `corsika7xxx` by the correct name of the compiled CORSIKA binary. `RUN000001.inp` is the name of the steering card file provided as an example. The radio part of the simulation run is configured with the files `SIM000001.reas` and `SIM000001.list` which are also provided as an example.

The example gnuplot script `SIM000001.gnu` will plot some simulation results, try it out.

## 7 Setting up your own simulations

In this section, we provide a short introduction on how to set up a full chain of CORSIKA simulations with CoREAS enabled.

### 7.1 Setup of the CORSIKA steering card file

Running a CORSIKA simulation with CoREAS is not much different from setting up any standard CORSIKA simulation. There are, however, some parameters of the CORSIKA steering card file that need special attention.

#### 7.1.1 Singular values

Some of the parameters can be specified as a range in CORSIKA. For simulations with CoREAS, these, however, should be set to specific values (and not a range). This applies to the keywords `ERANGE`, `THETAP` and `PHIP`. In practice, for these keywords you should specify the same value twice, e.g., `THETAP 30.0 30.0` for an air shower with 30 degrees zenith angle.

#### 7.1.2 Number of showers

CoREAS is only compatible with running one shower per CORSIKA run. This is done by setting `NSHOW 1` in your CORSIKA steering file.

#### 7.1.3 Energy cutoffs

The recommended energy cutoff for electrons/positrons is 401 keV. (This is motivated by technical and physics aspects and will ensure correct results with good performance.) A corresponding entry would be `ECUTS 3.000E-01 3.000E-01 4.010E-04 4.010E-04`.

#### 7.1.4 Thinning

Thinning can be used as usual. As a conservative guideline,  $10^{-6}$  thinning with weight limitation [8] produces very high-quality radio simulations.

#### 7.1.5 Observation level

You need to specify an observation level, which sets the height (in cm asl) to which CORSIKA traces the air shower. Naturally, this level should be at least as low as the observation level for which you want to calculate the radio emission. In fact, for inclined air showers, you should set the observation level in CORSIKA lower than the observation level of your radio antennas, maybe

even to negative values. (Otherwise, there is a region "above the shower" that has not been traced with CORSIKA but will be relevant for radio emission.)

### 7.1.6 Output directory

If you use the keyword *DIRECT* to write output to a directory other than the one where the CORSIKA binary resides, please make sure the directory path ends with a */*.

## 7.2 Setup of the CoREAS configuration files

You need two files to configure the CoREAS part of the simulation:

- *SIMxxxxxx.reas*: a file setting up the radio simulation parameters
- *SIMxxxxxx.list*: a file setting up the active observer (i.e., radio antenna) positions, see below for more details

Please note that the names of these two files are not flexible. They have to be called as stated above, with *xxxxxx* being the run number with leading zeroes. For run number 1, the names would thus be *SIM000001.reas* and *SIM000001.list*. Also, these files need to be placed in the directory which was specified in the *DIRECT* keyword in the CORSIKA steering card file.

### 7.2.1 Setup of the .reas file

Inside the *.reas* file, parameters are set in the manner

```
KeyWord=Value ; comment
```

where everything as of the semicolon is treated as a comment. In addition, lines starting with a *#* are treated as pure comments. Here, we give a short summary of the parameters that you can set in the *.reas* file.

#### CoreCoordinateNorth

This parameter sets the north coordinate of the air shower core position. Use the same coordinate origin as in the *.list* file!

#### CoreCoordinateWest

This parameter sets the west coordinate of the air shower core position. Use the same coordinate origin as in the *.list* file!

#### CoreCoordinateVertical

This parameter sets the vertical coordinate of the air shower core position. Use the same coordinate origin as in the *.list* file!

#### TimeLowerBoundary

Sets a global lower bound for the time window to be calculated. (Only applicable if *AutomaticTimeBoundaries=0* and not recommended, setting of *AutomaticTimeBoundaries* is preferred.) Value is in seconds, where 0 denotes the time when an imaginary leading particle propagating at the speed of light hits the specified shower core.

### **TimeUpperBoundary**

Sets a global upper bound for the time window to be calculated. (Only applicable if *AutomaticTimeBoundaries*=0 and not recommended, setting of *AutomaticTimeBoundaries* is preferred.) Value is in seconds, where 0 denotes the time when an imaginary leading particle propagating at the speed of light hits the specified shower core.

### **TimeResolution**

Sets the sampling resolution in seconds for the calculation of the time series data.

### **GroundLevelRefractiveIndex**

Specifies the refractive index at 0 m asl. The default value is set to 1.000292. If you want to switch off refractive index effects, set this to 1.0.

### **AutomaticTimeBoundaries**

If set to 0, the boundaries of the time windows are set globally by *TimeLowerBoundary* and *TimeUpperBoundary*. Otherwise, sets a time window in seconds for the calculation of the time series data that the code will position adequately for each individual observer. This decreases the amount of RAM needed for the simulation. A reasonable value is  $4e-07$  in combination with *ResolutionReductionScale*=5000.

### **ResolutionReductionScale**

If set to 0, all observers use the same sampling time resolution. Otherwise, sets a radial distance scale in cm on which the time resolution is repeatedly lowered, at the same time enlarging the time windows correspondingly. Activating this option significantly decreases RAM usage, but the output files will not have a common sampling time scale. If you can live with a variable sampling rate for different observers, a recommended value is 5000 in combination with *AutomaticTimeBoundaries*= $4e-07$ .

### **Offline-related keywords**

A number of keywords keep track of meta-information such as the CORSIKA steering card file name associated with the simulation or event numbers and GPS timestamps. These are in particular used by the Auger Offline software. These keywords do not influence the CoREAS simulation itself.

### **Purely informative keywords**

A number of keywords are written back to the .reas file after the end of the CORSIKA run for your convenience, but they are not used as input values for a given simulation. This includes things like the depth of shower maximum, distance of shower maximum, and the parameters describing the setup of the shower and the magnetic field.

### **Discontinued Keywords**

As you might have noticed, many keywords that were relevant for REAS3.x are no longer needed and have been removed.

It is recommended that you use the example file provided as part of this documentation as a starting point.

### 7.2.2 Setup of the .list file

To configure the observer locations, you have to provide them in the *.list* file. The syntax is like this:

```
AntennaPosition = 10000      0  140000  pole_100m_0deg
AntennaPosition =      0 -10000  140000  pole_100m_270deg
AntennaPosition = 40000      0  140000  pole_400m_0deg
AntennaPosition =      0 -40000  140000  pole_400m_270deg
```

Each line denotes an antenna position. The columns signify the position to north, the position to west and the height asl, all in cm, followed by a unique name for the observer. The coordinate origin is arbitrary, but must be the same as the one used in the *CoreCoordinateNorth*, *CoreCoordinateWest* and *CoreCoordinateVertical* statements specified in the *.reas* file.

### 7.3 Running CORSIKA simulations with CoREAS

Having prepared your CORSIKA steering card file (e.g. called *RUNxxxxxx.inp*) and the *SIMxxxxxx.reas* and *SIMxxxxxx.list* files to configure CoREAS, you are ready to run a simulation.

Run the CORSIKA binary as usual with input from your CORSIKA steering card file. You need to be in the directory with the CORSIKA binary to start the run. You should divert the on-screen output to a file for archiving purposes, e.g., *RUNxxxxxx.log*. A typical command would be:

```
./corsika-75xxx-linux </home/user/sim/RUN000001.inp
>/home/user/sim/RUN000001.log
```

If you get error messages, please make sure you refer to the CORSIKA user's manual for CORSIKA-related aspects. Once the simulation has finished, you will have a number of output files created by CORSIKA and CoREAS.

### 7.4 CoREAS output files

A number of files are written to disk. All data are saved as ASCII-text. The data compress very well if you have to save disk space. For an example with a run-number of 1, we get:

- *SIM000001.reas*: The input file is written back to disk after the simulation with the values that were actually used in the simulation run. In particular, any values imported from CORSIKA will be written to the corresponding fields in the *.reas* file.
- *SIM000001\_coreas.bins*: This file lists the files in the *SIM000001\_coreas/* directory together with the x, y and z coordinates of the corresponding observers. The next column has a universal value of zero for compatibility with older software. The last column denotes the axis distance of each observer location.
- *SIM000001\_coreas/*: This is the directory in which the main results (the time-series data for the individual ground bins) are saved.



The raw time-series data is then contained in one *raw\_antenna-id.dat* file per observer where the antenna-id was given in the *.list* file entries. The columns in the file denote the absolute time stamp and the north-, west-, and vertical component of the electric field.

All quantities listed in CoREAS output files are in cgs units! Thus, once you know what kind of quantity (e.g., field strength, distance, time) is listed, there remain no ambiguities.

The raw time-series data represent pulses calculated for infinite bandwidth. To filter these data with a filter and generate useful plots these raw data have to be reduced further.

## 8 Conventions

This section gives an overview of the conventions used by the CoREAS code.

### 8.1 Coordinates

Within the CoREAS code, the same coordinate conventions as in CORSIKA are being used. For spatial coordinates, this means that a right-handed coordinate system of x, y and z is used where x denotes the geomagnetic north direction, y denotes the west direction and z denotes the vertical direction. The same coordinate system applied to electric field vectors. For the azimuthal angles, 0 degrees denotes north, 90 degrees denotes west, i.e., counter-clockwise rotation.

Also, as in CORSIKA, an air shower is characterised by the direction into which it propagates — not the direction from which it is coming, as is usually done in experimental data. If an air shower has an azimuth angle of 0 degrees, this means that it is propagating to the north, i.e., coming from the south. If it has an azimuth angle of 90 degrees, this means that it is propagating to the west, i.e., coming from the east.

### 8.2 Units

Throughout the CoREAS code, cgs units are used. Exceptions are only made when importing values from other sources, and there should be a comment in the source code in these contexts.

All values being written out in data files by CoREAS are in cgs units.

## 9 Short guide for using MPI-parallelization

CORSIKA and CoREAS can be used with MPI to run parallelized simulations. In this mode, sub-showers are distributed to multiple cores, and the results are collected at the end of the simulation. This is particularly useful when your individual jobs become impractically long, in particular when they come close to wall-times enforced on a given computing cluster. There is obviously some overhead in bookkeeping, so if you want to run a library of many “short” jobs, MPI parallelization is not useful as it will increase the total required computing time.

In the following, I give a short guide on how to get MPI running on your local machine so that you can use the multiple cores provided by your CPU.

Of course, MPI can also use cores distributed over different computers, but the configuration of such an environment is beyond the scope of this manual.

There are several different versions of MPI around. One that requires no configuration and thus is particularly easy to get started with is *open-mpi*. On a current Ubuntu you can install it by:

```
sudo apt install openmpi-bin openmpi-common libopenmpi-dev
```

Before compiling CORSIKA with MPI, you need to set some environment variables. In the shell where you will run *coconut*, and from the CORSIKA main directory, run the command

```
source src/parallel/unsetoptflags.sh
```

Afterwards, run *coconut* as usual and in addition to the usual options select

```
    b - PARALLEL treatment of subshowers (includes LPM)
```

followed by

```
    2 - Library to be used with MPI system
```

Let *coconut* compile CORSIKA. In the *run* directory, you will find a binary called *mpi\_corsika...parallel\_runner*. You do not call this binary directly, but use the wrapper *mpirun* to call it.

Before you do so, however, you need to add a line configuring the parallel computation to your CORSIKA steering file. It looks like:

```
PARALLEL 1000 1.E5 1 T
```

The first parameter configures below which energy (in GeV) no parallelization of sub-showers is done any more. You can leave it at the standard value.

The second parameter is the energy of a sub-shower (in GeV) above which it will be split into separate parallel jobs. The number that you set here should have a reasonable value with respect to both the primary energy and the number of cores you want to run on. Say you want to run on 10 cores, then this number obviously should be at least a factor of 10 lower than your primary energy. To ensure a homogeneous distribution of the workload onto the cores, I suggest to apply an additional factor of 100. So for 10 cores, this number should be a factor of 1000 lower than your primary energy, i.e., the value of 1.E5 is reasonable for the example shower with primary energy of 1.E8 GeV. Setting this number much lower is not recommended, because the overheads in communication will increase, and the computation will become inefficient.

The last two parameters should be left unchanged, in particular the last one must be *T* for parallelization to work properly.

To run your simulation, then invoke the command

```
mpirun -c 10 ./mpi\_corsika75xxx-linux-coreas\_parallel\_runner  
/home/user/sim/RUN000001.inp  
>/home/user/sim/RUN000001.log
```

The number provided with the parameter *c* denotes the number of cores to run on. Be aware that one core serves as the “master” for book-keeping and is not available for shower simulations. Also please note that in contrast to the invocation of standard CORSIKA, the input file is provided as a true parameter, so without the < sign.

Please also check the *MPI-Runner\_GUIDE* provided in the *doc* directory for further information.

## 10 Example parameter files

As a basis for running simulations with CoREAS, we provide a set of example files here. They are also included in the CoREAS source code package in the subdirectory *doc/examples*.

### 10.1 CORSIKA

A suitable parameter file *RUN000001.inp* for running a  $10^{17}$  eV air shower looks like this:

```
RUNNR 1
EVTNR 1
SEED 1 0 0
SEED 2 0 0
SEED 3 0 0
PRMPAR 14
ERANGE 1.000E+8 1.000E+8
ESLOPE 0.000E+00
THETAP 0. 0.
PHIP 0.000E+00 0.000E+00
ECUTS 3.000E-01 3.000E-01 4.010E-04 4.010E-04
ELMFLG T T
THIN 1.000E-06 1.000E+02 0.000E+00
THINH 1.000E+00 1.000E+02
NSHOW 1
USER huege
HOST iklxds69
DIRECT './'
OBSLEV 140000.0
ECTMAP 1.000E+05
STEPFC 1.000E+00
MUMULT T
MUADDI T
PAROUT F F
MAXPRT 1
MAGNET 18.37 -13.84
LONGI T 5. T T
RADNKG 5.000E+05
DATBAS F
EXIT
```

### 10.2 CoREAS

The following is a parameter file *event.reas* to simulate radio emission for this CORSIKA simulation. (Comments have been removed here to improve readability, but are included in the electronic version.)

```
# CoREAS V1

# parameters setting up the spatial observer configuration:
```

```

CoreCoordinateNorth = 0
CoreCoordinateWest = 0
CoreCoordinateVertical = 140000

# parameters setting up the temporal observer configuration:

TimeResolution = 2e-10
AutomaticTimeBoundaries = 4e-07
TimeLowerBoundary = -1
TimeUpperBoundary = 1
ResolutionReductionScale = 0

# parameters setting up the simulation functionality:
GroundLevelRefractiveIndex = 1.000292

# event information for Offline simulations:

EventNumber = -1
RunNumber = -1
GPSSecs = 0
GPSNanoSecs = 0
CoreEastingOffline = 0
CoreNorthingOffline = 0
CoreVerticalOffline = 0
OfflineCoordinateSystem = Reference
RotationAngleForMagfieldDeclination = 0
Comment =

# event information for your convenience, not used as input:

ShowerZenithAngle = 0
ShowerAzimuthAngle = 0
PrimaryParticleEnergy = 1e+17
PrimaryParticleType = 14
DepthOfShowerMaximum = -1
DistanceOfShowerMaximum = -1
MagneticFieldStrength = 0.2300005511
MagneticFieldInclinationAngle = -36.99445254
GeomagneticAngle = 126.9944525
CorsikaFilePath = ./
CorsikaParameterFile = RUN000001.inp

A corresponding ground file SIM000001.list would look like this:

AntennaPosition = 10000      0  140000  pole_100m_0deg
AntennaPosition =      0 -10000  140000  pole_100m_270deg
AntennaPosition = 40000      0  140000  pole_400m_0deg
AntennaPosition =      0 -40000  140000  pole_400m_270deg

```

## 11 License

CoREAS is available to every scientist free of charge, but may not be used for commercial or military applications. You may not distribute the program or parts of it to other interested persons, but instead are asked to refer them to the official CORSIKA webpage for information on how to obtain the most recent version of the source code. Only this way, we can keep an overview of who is working with the code and inform about bug fixes. If you publish results based on CoREAS simulations, please cite the appropriate references mentioned during program startup. For more detailed copyright information, please read the copyright notice in the source code itself. For further information, please contact: Tim Huege (tim.huege@kit.edu)

## Acknowledgements

I would very much like to thank Ralf Ulrich, Tanguy Pierog and Dieter Heck for their help with making CoREAS an integral option of the CORSIKA build system.

## References

- [1] T. Huege, M. Ludwig, C. W. James. Simulating radio emission from air showers with CoREAS *AIP Conf. Proc.*, 1535:128–131, 2013.
- [2] D. Heck, J. Knapp, J. N. Capdevielle, G. Schatz, and T. Thouw. CORSIKA: A Monte Carlo Code to Simulate Extensive Air Showers. FZKA Report 6019, Forschungszentrum Karlsruhe, 1998.
- [3] T. Huege and H. Falcke. Radio emission from cosmic ray air showers. Monte Carlo simulations. *Astronomy & Astrophysics*, 430:779–798, 2005.
- [4] T. Huege and H. Falcke. Radio emission from cosmic ray air showers: Simulation results and parametrization. *Astropart. Phys.*, 24:116, 2005.
- [5] T. Huege, M. Ludwig, O. Scholten, and K. D. de Vries. The convergence of EAS radio emission models and a detailed comparison of REAS3 and MGMR simulations. *NIM A*, 662:S179–S186, 2012.
- [6] T. Huege, R. Ulrich, and R. Engel. Monte Carlo simulations of geosynchrotron radio emission from CORSIKA-simulated air showers. *Astropart. Physics*, 27:392–405, 2007.
- [7] C. W. James, H. Falcke, T. Huege, and M. Ludwig. An ‘endpoint’ formulation for the calculation of electromagnetic radiation from charged particle motion. *Phys Rev. E*, 84:056602, 2011.
- [8] M. Kobal and Pierre Auger Collaboration. A thinning method using weight limitation for air-shower simulations. *Astroparticle Physics*, 15:259–273, June 2001.

- [9] M. Ludwig and T. Huege. REAS3: Monte Carlo simulations of radio emission from cosmic ray air showers using an "end-point" formalism. *Astropart. Phys.*, 34:438-446, 2011.