

Cosima – the Cosmic Simulator of MEGALib

Andreas Zoglauer, Georg Weidenspointner, Michelle Galloway, Steven E. Boggs, Cornelia B. Wunderer

Abstract—Cosima is the Geant4-based simulator of the medium-energy gamma-ray astronomy library MEGALib. Within the MEGALib framework, Cosima is designed to be a simple, fool-proof interface to Geant4 and to enable the simulation of most of the measurement scenarios encountered by X-ray and gamma-ray detectors in space and on Earth.

Index Terms—Simulations, Geant4, gamma ray detectors, X-ray detectors, astrophysics, MEGALib, Cosima.

I. INTRODUCTION

SIMULATIONS are an essential development and analysis tool for X-ray and gamma-ray detectors. During the design phase of such instruments, they allow determining the performance of the instruments with respect to the desired science objectives. In addition, they enable the optimization of the design by performing trade-off studies between variations of the detector setup, various instrument orbits, shielding thicknesses, optimization of the passive material with respect to detector activation, and many more. Finally, simulations help to prepare and to understand calibrations and measurements of the instrument. Overall, simulations are a tool for cost effective trade-off studies, they allow identifying and minimizing mission risks, and result in detectors optimized for the science goals of the mission. Thus realistic simulations are a key to modern space telescope development as well as for the development of X-ray and gamma-ray detectors for terrestrial applications such as nuclear monitoring and threat detection systems.

One of the tools capable of and designed to perform the above tasks is Cosima, the Geant4-based [1] cosmic simulator of MEGALib [2]. MEGALib, the Medium-Energy Gamma-ray Astronomy library, is a simulation and data analysis tool for hard X-ray and low-to-medium-energy gamma-ray telescopes (intended for energies from a few keV to hundreds of MeV) encompassing the complete chain from simulations to high-level data analysis such as image reconstruction. While its main application area is Compton telescopes in space, it can also be applied to terrestrial hard X-ray and gamma-ray detectors (ranging from simple spectrometers, via coded-mask systems to Compton and pair telescopes). MEGALib has a

completely object-oriented design and is written in C++. It is based on ROOT [3]. The main design goals of its Cosima subsystem are (1) to be a simple, fool-proof interface to Geant4, (2) to enable all measurement scenarios of hard X-ray and soft-to-medium-energy gamma-ray telescopes/cameras in space and on Earth, and (3) to be fully integrated into the MEGALib framework.

Cosima, along the whole MEGALib framework, was originally written as the simulation and data analysis tool for the combined Compton and pair telescope MEGA [4] (hence the name). The necessity to compare the performance of MEGA with other detector techniques and to estimate the performance of potential satellite missions based on the MEGA detector technology required a very flexible design of all parts of the suite (geometry description, simulation tool, detector effects engine, and analysis tools). This allowed MEGALib to be later applied to various other existing or envisioned space, balloon-borne and terrestrial detector systems, for example GRIPS [5], ACT [6], NCT [7], NuSTAR [8], COMPTEL [9], TGRS [10], HEMI [11] and many more.

For a general overview of MEGALib see [2] and [12]. Cosima (in a package with MEGALib) can be freely downloaded from the following webpage:

<http://www.mpe.mpg.de/MEGA/megalib.html>

The following sections will give an overview of the inputs, execution, and results of simulations with Cosima. It is neither intended as a replacement for the manual nor as a description of the implementation details, but rather as an overview of Cosima's capabilities, its usage, and its application areas.

II. THE COMPONENTS OF THE SIMULATOR

A. Overview

As part of MEGALib, Cosima has to utilize the same geometry and detector description as all other MEGALib programs, it has to provide an output compatible with all other MEGALib programs, and from the programming point of view reuse as many of the MEGALib core classes as possible.

Figure 1 shows an overview of the simulation and data analysis path within MEGALib. The first input to the simulator is a geometry description (volumes of certain shapes and materials at certain positions) as well as a detector description (e.g. a pixel detector with certain energy resolution, etc.). The second main input to the simulation is the source description (e.g. some point sources and background distributions for astrophysics) as well as some commands steering the simulation (e.g. Geant4 physics lists, output options, a stop criterion, etc.). With those inputs the simulation can be

Manuscript received November 13, 2009.

A. Zoglauer, S. E. Boggs, and M. Galloway are with the Space Sciences Laboratory, University of California at Berkeley, 7 Gauss Way, Berkeley, CA 94720 USA (e-mail: zog@ssl.berkeley.edu, boggs@ssl.berkeley.edu, shell@ssl.berkeley.edu)

G. Weidenspointner is with the Max-Planck-Institut für extraterrestrische Physik, Garching, Germany and with the MPI Halbleiterlabor, München Germany (e-mail: georg.weidenspointner@hll.mpg.de)

C. B. Wunderer was with the Space Sciences Laboratory, University of California at Berkeley, Berkeley, CA 94720 USA

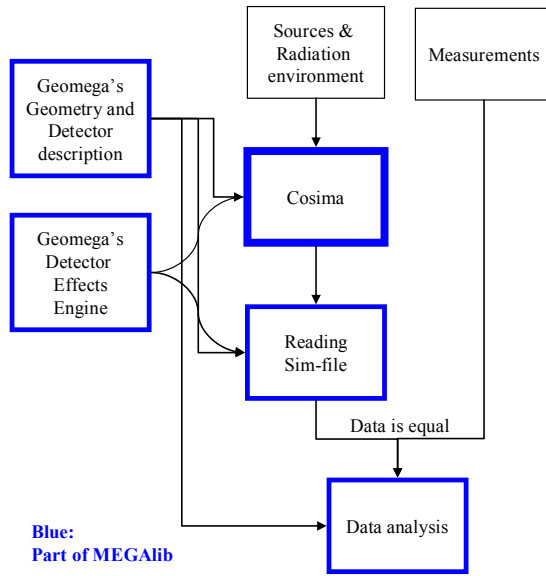


Figure 1: The simulation data flow

started. From the users point of view Cosima is designed to be more or less a black box – the user does not need to understand the details of the simulation. Most of the detector effects are not applied to the simulated data within Cosima (with exception of voxelization and charge/light loss) but are applied after the simulation, when the simulation file is read in with MEGALib. After this stage the simulated data is completely identical to real data and the high-level analysis tools of MEGALib (e.g. event reconstruction, image reconstruction, response analysis, etc.) can be applied.

B. Geomega – geometry and detector description

Geomega (“Geometry for MEGALib”) is the universal geometry and detector description library of MEGALib. All information is stored and retrieved from a simple ASCII file. An excerpt of a typical geometry and detector setup file can be found in Figure 2. The file is keyword based and reflects the object-oriented structure of Geomega. For example, the volume-object “Wafer” (see Figure 2) consists of the parameters shape, material, position, rotation, mother volume, visualization attributes, etc.

The key sections of a geometry file are:

1. a global section (name, version, included files, constants, etc.)
2. a list of all used materials and their parameters
3. volumes (including copies of volumes) and their placement
4. the detector types and their characteristics defining the detector effects engine
5. trigger criteria

In addition, it is possible to split the geometry over several files and simply include those files in a master geometry file. Moreover, constants can be declared to repeatedly use certain

```

Name SimpleGermaniumDetector

<cut>

// Material description
Material Germanium
Germanium.Density      5.323
Germanium.Component    72.59 32.0 1

<cut>

// Volume description
Volume Wafer
Wafer.Color            4
Wafer.Visibility       1
Wafer.Material         Germanium
Wafer.Shape            BOX 4.0 4.0 1.0

Constant ZStart { -ZDist*(NWafers-1) }
Constant ZDist   3.0
Constant NWafers 5

For I NWafers ZStart ZDist
  Wafer.Copy Wafer_%I
  Wafer_%I.Position 0.0 0.0 $I
  Wafer_%I.Mother   World
Done

<cut>

// Detector description
Strip3D GeD
GeD.DetectorVolume      Wafer
GeD.SensitiveVolume     Wafer
GeD.StructuralPitch     0.0 0.0 0.0
GeD.StructuralOffset    0.0 0.0 ZDist
GeD.Offset              0.2 0.2
GeD.StripNumber         38 38
GeD.NoiseThreshold      15
GeD.TriggerThreshold    30
GeD.DepthResolution     100 0.2
GeD.EnergyResolution    10 0.417
GeD.EnergyResolution    20 0.425

< cut >

// Trigger: at least one hit in one Ge detector
Trigger T1
T1.Veto false
T1.TriggerByChannel true
T1.Detector GeD 1

```

Figure 2: Excerpt of a typical geometry file

numbers or to easily find and change them in the setup file. Furthermore, Geomega is capable of performing mathematical operations (“{ }” brackets in the example), execute for-loops (in the example %I is the current index running from 1 to NWafers and \$I accesses the current value starting from ZStart and being incremented by ZDist), if-conditions, and a few more. For a complete list of all features in the latest version consult the Geomega manual.

Geomega itself consists of two parts, the library and its graphical user interface. The GUI, which accesses only parts of the library, is mostly used for geometry visualization, determination of masses, listing of the geometry tree, converting the geometry into the MGGPOD [13] format, and a few more. Figure 5 shows such a visualization of a geometry using ROOT’s OpenGL viewer. The library part of Geomega

is used by almost all other programs of MEGALib. It is a set of classes encapsulating all the required functionality arranged around the central class MDGeometry. For example the volume information is stored in the class MDVolume, its shape is stored in classes derived from MDShape (e.g. MDShapeTUBE), the detector information in classes derived from MDDetector (e.g. MDStrip2D), etc. All the information is organized in and accessed from the main classes MDGeometry and MDGeometryQuest, which provide the high-level functionality such as access to the ROOT geometry display, calculation of absorption probabilities, volume sequences for individual interaction positions, and many more.

A key element of the library is the detector effects engine. Geomega contains several detector types with special characteristics. These include 2D strip detectors (e.g. MEGA Silicon wafers), 3D strip detectors with depth resolution (e.g. NCT Germanium detectors), voxel detectors, MEGA-like calorimeters, COMPTTEL-like Anger cameras, and simple one volume detectors without position resolution. The parameters given to the detectors steer their detector effects engines. In general, these include:

- Not detectable energy (e.g. due to charge trapping) in the form of a fractional energy loss on a 3D-grid representing the detector volume
- Voxelization (i.e. 2D/3D centering of measured positions) of individual energy deposits into the strips/pixels/voxels/guardrings of the detector
- Strip/pixel failure
- Energy resolutions (Gaussian, Lorentz, Gauss-Landau) and shifts as a function of the measured energy
- ADC (energy) overflows
- Noise thresholds: all summed energy deposits in one strip/pixel/voxel below this threshold are assumed not to be distinguishable from detector noise and thus not read out
- Time resolution of individual hits (assuming the detector is capable of time measurements)
- Complex trigger criteria including trigger thresholds

The first two, not detectable energy and voxelization, are already performed in Cosima during the simulations in order to minimize the size of the stored data file. However, in order to allow the user to write a custom detector effects engine, all the data can be written to a file without voxelization. All later stages of the detector effects engine are performed during reading of the generated simulation file (output of Cosima) in order to allow the user to modify the detector characteristics without having to re-run the simulations again which is usually the most time consuming part of the analysis.

For a full list of all features of Geomega and its detector effects engine consult the manual and look at the examples provided within MEGALib.

```
Version          1

# Geometry section
Geometry         Satellite.geo.setup

# Physics section
PhysicsListEM    Penelope
PhysicsListHD    QGSP-BIC-HP

# Output format section
StoreSimulationInfo  true
DiscretizeHits      true

# Defining the run
Run SpaceSim
SpaceSim.FileName   CrabObservation
SpaceSim.Time       10000.0

# Defining the runs trigger criterion
SpaceSim.Trigger D1D2
D1D2.AddPositiveDetectorType 1
D1D2.AddPositiveDetectorType 2

# Defining a point-source source
SpaceSim.Source Crab
Crab.ParticleType 1
Crab.Beam FarFieldPointSource 0 0
Crab.Spectrum PowerLaw 800 100000 2.17
Crab.Flux 0.004972 // ph/cm2/s
Crab.Polarization 1.0 0.0 1.0 0.0 0.4

# Defining a second source
SpaceSim.Source CosmicPhoton
CosmicPhoton.ParticleType 1
CosmicPhoton.Beam FarFieldAreaSource 0.0
113.4 0.0 360.0
CosmicPhoton.Spectrum File
CosmicPhotonSpectrum.dat
CosmicPhoton.Flux 0.035984 // ph/cm2/s
```

Figure 3: A typical source file

C. Defining the simulation input for Cosima

The simulation is steered by a simple ASCII input file (called “source file”), which is passed to Cosima upon the program start. In order to simplify the interface, and to make the simulation more fool proof, the options the user has access to are limited to the minimum necessary and those proven reasonable for the simulation in the given astrophysical and terrestrial measurement environment. For normal simulations, Geant4’s default macro processing is not enabled (however it can be used for visualization tasks).

A typical source file (see for example Fig. 3) contains the following main sections: geometry, physics, output format, runs and sources.

The first input is the geometry file. Internally Cosima first generates a Geomega geometry and then translates the parts relevant for the simulation into Geant4 materials, volumes, placements, and sensitive detectors.

The physics section describes the requested physics lists. The user can choose between various low-energy electromagnetic lists e.g. Livermore and Penelope, and between

QGSB-BIC-HP and QGSB-BERT-HP for hadronic simulations. These hadronic physics lists have been proven to give the best results for prompt proton interaction and detector activation simulations in the astrophysical case. Additional options include the simulation of fluorescence, and the handling of generated radioactive isotopes and their decays for activation simulations of telescopes in the space radiation environment.

The file output section defines the information content and style of the output file of Cosima, e.g. whether or not information about the simulation history is stored, whether or not the hits are discretized into detector strips/pixels, whether or not ionization information is stored, the number of significant digits in the output file, etc.

The next section defines a simulation run. The file might contain multiple runs which are processed in sequence. A run has to contain an output file name, a stop criterion (observation time, triggered events, or started events), a trigger criterion (simple positive criterion without vetoes and thresholds such as at least one hit in the strip detector and one in the calorimeter), and one or more sources. A source is defined at least by a particle type, a beam, a spectrum, and an average total flux integrated over beam and spectrum.

The various available beam types include point sources, area sources, and zenith-dependent distributions at infinite distance (far field). For the near field it includes point sources (beamed, diffracted), line, box, sphere, and disk sources, various beams (homogenous profile, radial profile, and those with a profile defined by file), cone beams (flat and Gaussian profiles), illuminated disks and boxes, emission patterns loaded from a file, as well as homogeneous emission from a volume of the geometry (e.g. in combination with radioactive isotopes).

The spectral types include mono-energetic, linear, power law, broken power law, Gaussian, black-body, and thermal Bremsstrahlung distributions as well as those loaded from file.

For more details, a full list of all features and all restrictions see the latest version of the Cosima manual.

D. The simulation process

The simulation is started by simply calling the main program via “cosima MyInputFile.dat”. By default a random seed for Geant4’s random number generator is used, but the user can give a special seed via the command line (“-s”). Other command line options include setting the verbosity of Cosima (and thus Geant4), executing only a specific run from the source file, etc.

Cosima is set up as a standard Geant4 user application following the principles laid out in Geant4’s application development manual. After parsing the command line options and the source file, Cosima initializes its run manager (*MCRunManager*). It is responsible for processing all of the runs defined in the source file in sequence. Afterwards, the chosen physics lists are initialized, followed by the event action (*MCEventAction*), where the trigger options, stop

criteria, and output file options are set up. Then the detector construction is called (*MCDetectorConstruction*). It first initializes a Geomega geometry and then converts all relevant elements into a Geant4 detector setup. For each detector type (e.g. strip detector, calorimeter, etc.) an appropriate sensitive detector object (e.g. *MC2DStripSD* derived from *G4VSensitiveDetector*) and hit object (e.g. *MC2DStripHit* derived from *G4VHit*) is created. Afterwards the primary generator action class (*MCPPrimaryGeneratorAction*) is initialized, which in turn initializes the source objects (*MCSource*) belonging to the current run (*MCRun*). These sources contain not only all source parameters, but are also capable of generating the start parameters of the simulated particle (type, position, direction, energy, polarization) and transferring them to a *G4GeneralParticleSource* belonging to the *MCPPrimaryGeneratorAction* class for simulation. Finally, Cosima has its own stepping action (derived from *G4SteppingAction*) and tracking action (*G4TrackingAction*) classes, which enable the storage of the particle history in the output file.

As in each standard Geant4 application, the simulation is started with the run manager *BeamOn()* function. When during the simulation the *PrimaryGeneratorAction* is called, each of the available sources is scanned to determine at which observation time the next event is to occur: at the end of the initialization of the last particle in the source (or during initialization of the object) the observation time of the next particle of the source has been randomly determined from the known flux/intensity of the source (using a Poisson distribution). After advancing the time, for the selected source, the start parameters are randomly determined from the emission profile, energy spectrum, emission direction, etc. Those parameters are handed to the *G4GeneralParticleSource* for simulation. During the simulation the particle track information is intercepted in the *MCSteppingAction* and *MCTrackingAction* classes, and transferred to the *MCEventAction* class for later storage in the output file. If there are energy deposits in the sensitive detectors, those are handled in the appropriate sensitive detector classes. For example hits in pixel detectors are checked if they occur in the guard ring or the sensitive detector material, and either a guard-ring hit or a pixel hit is created. The position of the hit is centered in the pixel. At the end of the simulation of the event, the *MCEventAction* class checks for triggers, dumps the event information in the chosen format into the simulation file, and finally checks if the stop criterion for the run is fulfilled.

E. The output of Cosima

A typical output of Cosima can be found in Figure 4. Besides some general information such as the ID of the event and the observation time, it contains two main sections, the IA (interaction) and the HT (hit) section.

The IA section describes the complete interaction history of the primary particle and all secondaries. The details of the

```

SE
ID 4 568
TI 5.535837965
< cut >
IA INIT 1; 0;0;0.000000000000e+00; 12.35804;
11.28213; 40.00000;0; 0.00000; 0.00000;
0.00000; 0.00000; 0.00000; 0.00000;
0.000;1;-0.00000;-0.00000;-1.00000;
0.00000; 0.00000; 0.00000; 805.798
IA COMP 2; 1;1;2.034125846877e-09; 12.35804;
11.28213; -20.98156;1;-0.04567; 0.10502;-
0.99342; 0.00000; 0.00000; 0.00000;
797.722;3; 0.39452;-0.90714;-0.14644;
0.00000; 0.00000; 0.00000; 7.977
IA PHOT 3; 1;2;2.690040161265e-09; 11.45989;
13.34728; -40.51599;1; 0.00000; 0.00000;
0.00000; 0.00000; 0.00000; 0.00000;
0.000;3;-0.58865; 0.35609;-0.72574;
0.00000; 0.00000; 0.00000; 764.558
< cut >
HTsim 1; 12.34950; 11.26850; 21.00000; 8.07601;
0.0e+00; 2
HTsim 2; 11.32000; 13.48000; -40.53665;
628.02789; 0.0e+00; 5;6;7;8;9
HTsim 2; 10.78000; 12.94000; -41.74642; 7.76044;
0.0e+00;10
HTsim 2; 10.24000; 13.48000; -42.25238;
161.93403; 0.0e+00; 11;12;13

```

Figure 4: Excerpt from a typical simulation file

output can be set in the source file. One IA line contains the interaction process (e.g. COMP for Compton), the ID of this interaction and its predecessor interaction, the interaction position and time, as well as the new direction, polarization and energy of the original and secondary particle (if any). In later parts of MEGALib this information is used for simulation diagnostics as well as for the generation of detector response files for event reconstruction and imaging.

The HT section describes the hits in the detectors, such as the detector ID, position in world coordinates, time (in the above case, the detector is not capable of measuring time, hence the “0.0e+00” as 6th number), and energy. Depending on the detail chosen for the IA section, it is followed by the IDs of the IAs which have contributed to this hit.

In order to easily analyze the IA and HT section of the simulation file, MEGALib contains a library which rebuilds the interaction sequence tree, called Sivan (Simulated Event Analysis). This enables an easy comparison between the reconstructed path of the photon and its main interaction process as well as the simulated path and processes. This knowledge is required for generating response files for event and image reconstruction.

While at this stage the positions are already centered in the voxels of the detectors, the full detector effects engine has not yet been applied (e.g. noise, thresholds, etc). This happens the moment the simulation file is read in within MEGALib (e.g. at the beginning of Compton event reconstruction). At this stage the simulated events should have the same characteristics as real, measured events. Then all the high level data analysis tools of MEGALib can be applied such as Compton or pair event reconstruction, image reconstruction, etc.

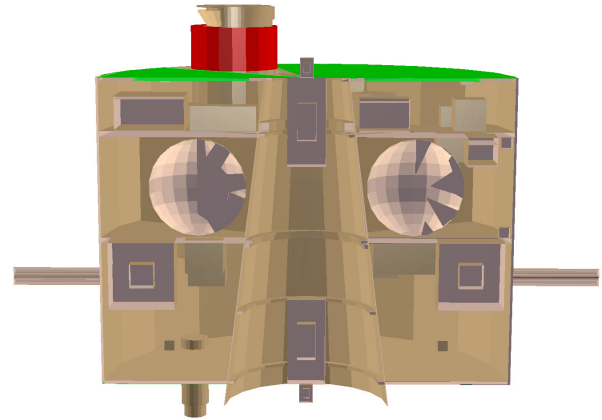


Figure 5: Geomega simulation model of the Wind spacecraft and the TGRS detector (top-left structure)

III. EXAMPLES

A. Simulations of the TGRS detector in interplanetary space

TGRS, the Transient Gamma-Ray Spectrometer, is a Germanium detector aboard the Wind spacecraft [10]. It was designed to detect gamma-rays from ~25 keV up to ~8 MeV. The simulation geometry can be found in Figure 5. For the detector effects engine, a simple one volume detector was assumed, with an energy resolution as measured with the real detector and a trigger threshold of ~10 keV.

TGRS measurements have been performed mostly in interplanetary space. There, the detector was mainly exposed to cosmic photons and protons. The simulation of TGRS consists of 4 parts. The first part is the simulation of the cosmic photon component. The emission distribution was isotropic, and the spectrum was taken from [14]. The second component is the prompt cosmic protons component. Its spectrum was calculated for early 1995 (the date of the measurements) using the GALPROP Galactic cosmic-ray

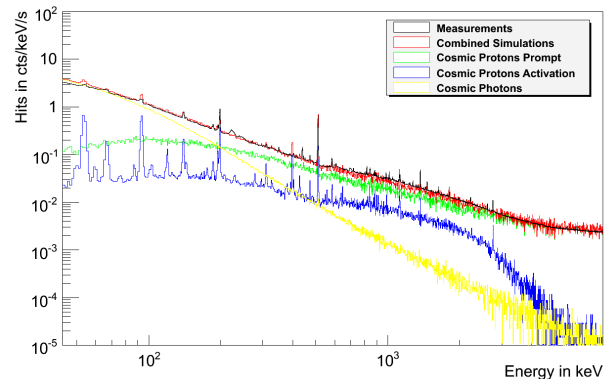


Figure 6: Comparison between measurements (black) and simulations (red: combined simulation) of the TGRS detector

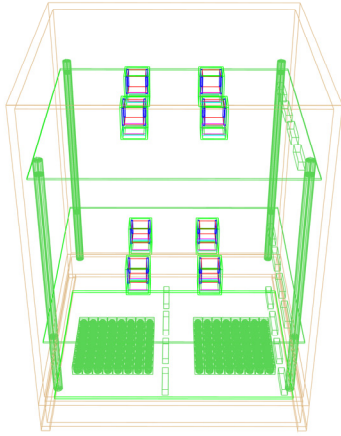


Figure 7: Mass model of the HEMI test setup for simulations

propagation code [15]. In order to perform activation simulations, the prompt cosmic proton component was simulated in a special Cosima mode, the activation build-up mode. It stores the generated isotopes, their excitation state, and the volume in which they have been generated in a file. During the third step of the simulations (which does not perform actual Monte Carlo simulations) this information is used to determine the activation after a given time in orbit using Cosima, and write the decays per isotope, volume and time into an activation file. In the final step this information is used as sources for the final simulation of radioactive decays.

Figure 6 shows a comparison between measurements (black) and simulations (red: combined simulations). While Cosima is capable of reproducing the overall shape quite nicely (considering the remaining uncertainties of the cosmic proton input, not simulated cosmic alpha particles, and remaining uncertainties in the geometry), there are some differences in the line strengths, which are a known problem in Geant4 [16].

For more details about the TGRS simulation setup and results see [17]. In addition, a similar comparison with the MGGPOD simulation suite has been performed in [13].

B. Simulations of the HEMI test setup

HEMI, the high efficiency multi-mode imager, is a combined coded-mask and Compton camera [17]. It consists of 1 cm^3 CZT coplanar-grid detectors. Its initial measurement test setup contained 8 CZT detectors arranged in 2 layers with 4 detectors each. The test setup is only capable of measuring Compton events. Figure 7 shows the Geomega geometry of the test setup.

The detector effects engine of HEMI includes a charge loss setup derived from charge transport simulations, a Gauss-Landau-shaped energy resolution function, which mimics the typical low-energy tails of CZT detectors, and finally an energy calibration which corrects the initial energy loss.

Figure 8 shows a comparison of the angular resolution measure (ARM) of HEMI between simulations and measurements of a ^{137}Cs source on axis at 50 cm distance. The

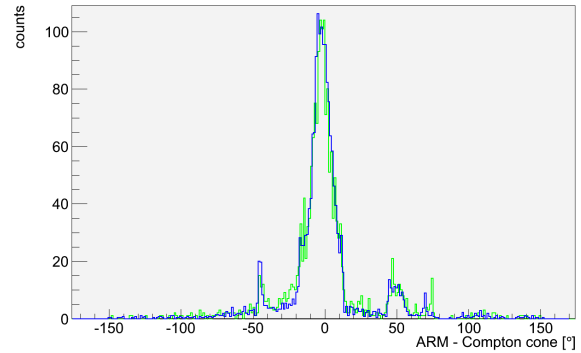


Figure 8: The angular resolution measure of a ^{137}Cs at 50 cm distance on-axis. The measurements are in blue, the simulations in green.

ARM shows a distribution of the smallest angular distance between the Compton cone and the known origin of the gamma ray. In order to achieve an agreement of the measured and the simulated ARM all elements of the simulation pipeline need to work properly, including the geometry, the Geant4 simulation, and all elements of the detector effects engine (charge-loss, energy resolution, calibration, thresholds, etc). Therefore, the agreement achieved in this figure shows nicely the end-to-end performance of the Cosima/MEGALib simulation setup.

For more details about measurements and simulations with HEMI see [11] and [17].

IV. SUMMARY

Cosima is the simple and fool-proof interface to Geant4 within MEGALib. It allows the simulation many of the measurement scenarios which are encountered by hard X-ray and soft-to-medium-energy gamma-ray detectors in space and ground applications.

ACKNOWLEDGMENT

We would like to thank all people who have contributed to Cosima and Geomega over the past years with code, suggestions for improvements and bug reports.

REFERENCES

- [1] S. Agnostelli et al., "Geant4 – a simulation toolkit", *NIM A*, vol. 506, pp. 270-278, 2003
- [2] A. Zoglauer, R. Andritschke, F. Schopper, "MEGALib – the Medium Energy Gamma-ray Astronomy Library", *New Astronomy Reviews*, vol. 50, no. 7-8, pp. 629-632, 2006
- [3] R. Brun, F. Rademakers, "ROOT – An Object Oriented Data Analysis Framework", *NIM A*, vol. 389, pp. 81-86, 1997
- [4] A. Zoglauer, "First Light for the next Generation of Compton and Pair telescopes", Doctoral thesis, TU Munich, 2005
- [5] J. Greiner et al., "Gamma-ray Burst Investigation via Polarimetry and Spectroscopy (GRIPS)", *Experimental Astronomy*, vol. 23, no. 1, pp. 91-120, 2009
- [6] S.E. Boggs, "The Advanced Compton Telescope mission", *New Astronomy Reviews*, vol. 50, no. 7-8, pp. 604-607, 2006
- [7] S.E. Boggs et al., "Performance of the Nuclear Compton Telescope", *Experimental Astronomy*, vol. 20, no. 1-3, pp. 387-394, 2005

- [8] F.A. Harrison, "Development of the HEFT and NuSTAR focusing telescopes", *Experimental Astronomy*, vol. 20, no. 1-3, pp. 131-137, 2005
- [9] Schönfelder, V. et al., "Instrument description and performance of the imaging gamma-ray telescope COMPTEL aboard the Compton Gamma-Ray Observatory", *The Astrophysical Journal Supplement Series*, vol. 86, pp. 657-692, 1993
- [10] A. Owens, R. Baker, T. L. Cline, N. Gehrels, J. Jermakian, T. Nolan, R. Ramaty, H. Seifert, D. A. Shephard, G. Smith, D. E. Stilwell, B. J. Teegarden, C. P. Cork, D. A. Landis, P. N. Luke, N. W. Madden, D. Malone, R. H. Pehl, H. Yaver, K. Hurley, S. Mathias, and A. H. Post, Jr., "A High-Resolution GE Spectrometer for Gamma-Ray Burst Astronomy", *Space Science Reviews*, vol. 71, pp. 273-296, Feb. 1995.
- [11] A. Zoglauer, M. Galloway, M. Amman, S.E. Boggs, J.S. Lee, P.N. Luke, L. Mihailescu, K. Vetter, C.B. Wunderer, "First Results of the High Efficiency Multi-mode Imager HEMI", *Nuclear Science Symposium Conference Record, 2009 IEEE*
- [12] A. Zoglauer, R. Andritschke, S.E. Boggs, F. Schopper, G. Weidenspointner, C.B. Wunderer, "MEGALib – Simulation and Data Analysis for Low-to-medium-energy Gamma-ray Telescopes", *Proceedings of the SPIE*, vol. 7011, pp. 70113F, 2008
- [13] G. Weidenspointner, J.M. Harris, S. Sturmer, B.J. Teegarden, C. Ferguson, "MGGPOD – a Monte Carlo suite for Modeling Instrumental Line and Continuum Backgrounds in Gamma-Ray Astronomy", *The Astrophysical Journal Supplement Series*, vol. 145, no. 1, pp. 69-91, 2005
- [14] D. E. Gruber, J. L. Matteson, L. E. Peterson, and G. V. Jung, "The Spectrum of Diffuse Cosmic Hard X-Rays Measured with HEAO 1", *The Astrophysical Journal*, vol. 520, pp. 124-129, Jul. 1999.
- [15] I. V. Moskalenko, A. W. Strong, J. F. Ormes, and M. S. Potgieter, "Secondary Antiprotons and Propagation of Cosmic Rays in the Galaxy and Heliosphere", *The Astrophysical Journal*, vol. 565, pp. 280-296, Jan. 2002.
- [16] S. Hauf et al., "Progress with Geant4-based Radioactive Decay Simulations Using the example of Simbol-X and the IXO WFI", *Nuclear Science Symposium Conference Record, 2009 IEEE*
- [17] A. Zoglauer, G. Weidenspointner, C.B. Wunderer, S.E. Boggs, "Status of Instrumental Background Simulations for Gamma-ray Telescopes with Geant4", *Nuclear Science Symposium Conference Record, 2008 IEEE*
- [18] M. Amman, P.N. Luke, J.S. Lee, L. Mihailescu, K. Vetter, A. Zoglauer, C.B. Wunderer, M. Galloway, S.E. Boggs, H. Chen, P. Marthandam, S. Awadalla, S. Taherion, G. Bindley, "Detector Module Development for the High Efficiency Multi-mode Imager", *Nuclear Science Symposium Conference Record, 2009 IEEE*