ATLAS detector simulation: status and outlook

D. Costanzo
University of Sheffield (UK)

A. Dell'Acqua, A. Di Simone, M. Gallas, A. Nairz
CERN (Switzerland)

A. Rimoldi
INFN and Univ. of Pavia (Italy)

J. Boudreau, V. Tsulaia
Univ. of Pittsburgh (USA)

December 19, 2005

Abstract

The simulation program for the ATLAS experiment at CERN is currently in a full operational mode and integrated into the common analysis framework of ATLAS, Athena. The robustness of the application was proved during Data Challenge 2 (DC2). The simulation software, based on GEANT4, has been interfaced within Athena and to GEANT4 itself using the LCG dictionaries and Python scripting. The Python interface has added the flexibility, modularity and interactivity that the simulation tool needs to manage, in a common way, the different simulation setups: full ATLAS detector, test beams and cosmic ray studies. The comparison with real data has been possible in the context of the ATLAS Combined Test Beam (2004) and ongoing cosmic ray studies. This paper presents a report on the status of the simulation, together with an overview of the applications and the results of the validation efforts.

1 The Geant4 ATLAS detector simulation

Starting from 2004, the GEANT4 [2] simulation framework has replaced the previous GEANT3-based simulation program [3] as the only officially supported software for the ATLAS detector simulation. It consists of a full-featured, object-oriented simulation suite (G4ATLAS), integrated in the common framework for the ATLAS offline software (Athena) [4]. The implementation of G4ATLAS has been driven by key concepts like dynamic loading and action-on-demand. The resulting software is thus highly-configurable and flexible, and can therefore accommodate all the requirements the simulation of a complex experiment such as ATLAS calls for. The user can choose *at run time* which kind simulation to run (whether full detector, test beam setup or cosmic simulation). However, since the underlying simulation code is always the same, consistency is ensured throughout all the different applications.

1.1 Geometry description

The geometrical description of the ATLAS detector has been decoupled from the simulation framework. A dedicated set of classes (GeoModel) provides all the necessary volume and material description. The GeoModel description is automatically translated into a GEANT4 geometry at run time. On the other hand, being GeoModel completely independent from GEANT4, *the same description* can be used also by reconstruction applications, thus ensuring total agreement between the geometries used for simulation and reconstruction. The GeoModel description has been optimized to handle the large number of volumes (about 10^6) involved in the description of the ATLAS experiment: the number of volumes itself is kept to the minimum and, whenever possible, volume parameterization is used to improve the performance.

Figure 1 shows some 3D views obtained with geometry display tools (either from Geant4 or GeoModel). The different subdetectors are represented. The number of volumes involved in the GeoModel description of the ATLAS subdetectors is shown in Table 1.1

detector	number of volumes	comments
Pixel	6000	
SCT	40500	
TRT	300000	parameterized
LAr	142500	parameterized
Tile	80500	parameterized
Muon Chambers	451000	parameterized
Toroids	1000	

Table 1: Number of simulated volumes for each subdetector

Figure 1: Overview of the geometrical description used by the simulation. All the images shown here are obtained directly by Geant4 or GeoModel.

1.2 Job control

In the Athena framework, jobs are configured and controlled using the `python` scripting language: `python` commands are grouped in *jobOptions* files, which contain all the information about the job to run (e.g. which algorithms to run, what input files to use, the data to be written in the output file, *etc.*). The user has thus all the advantages from a full featured programming language, such as flexibility and ease-of-use, and can use the existing bindings to interact directly with the underlying C++ code. GEANT4, on the other hand, has no native interface to `python` and the configuration of a simulation job is normally achieved using specific commands, which can be grouped into *macro* files. For the simulation of a complex experiment such as ATLAS, the number of macro files involved increases rapidly, leading to maintainability issues. Moreover, the macro files mechanism does not integrate with Athena's `python` job configuration, resulting in an unacceptable lack of flexibility. To solve this problem, a `python` layer has been developed (PyG4Atlas) which allows, using bindings to the underlying C++ G4ATLAS code, to fully configure a simulation job by means of `python` commands. Integration in the Athena framework is thus also achieved for what concerns job configuration, while the efforts requested to the users to interact with the simulation are kept to a minimum.

Figure 2: Job control mechanism with and without the python interface to G4.

A schematical view of the job control achievable with and without PyG4Atlas is shown in Figure 2. In the old approach (without PyG4Atlas), the user configures the Athena job *directly* using a `jobOptions` file. The configuration of the simulation, however, must be done by means of macro files, which must be prepared beforehand by the user. Each macro file will configure properly one particular aspect of the Geant4 simulation, either acting directly on the Geant4 kernel or on the FADS framework. The list of macros to be used can be specified in the `jobOptions` file. As already mentioned, the main drawback of this approach is that the user has no direct control at run time on the Geant4 configuration.

The new configuration mechanism can be divided in four steps:

- The macro file interface is dropped and replaced by new simple C++ classes (G4AtlasControl for FADS configuration and G4AtlasInterface for Geant4 direct configuration).
- The public interfaces of the new classes are exported to python using the SEAL PyLCGDict mechanism.
- A new package (G4AtlasApps) provides a simulation engine, written in python, which uses the dictionaries created in the previous step.
- The user can now configure the simulation engine directly (also *interactively*) from the same jobOptions file used for Athena configuration, thus obtaining full control on the Geant4 configuration at run time.

2 Examples of applications

While G4ATLAS has been designed for the simulation of the whole ATLAS detector, its flexibility allows it to be used in a wide range of applications. Simulation of single subdetectors in custom positions is possible, and easily achievable using python scripts. Moreover, additional volumes can be easily added to the ATLAS detector, using either

Figure 3: View of the Combined Test Beam experimental setup. The picture was obtained with Geant4 geometry display tools. Only the ATLAS subdetectors are shown here. The simulation includes however also ancillary detectors and dead materials.

GeoModel or, for simple shapes, python scripts. We show in this section two applications (the 2004 Combined Test Beam and the Cosmic Setup) which fully exploit the flexibility and ease-of-use of G4ATLAS.

2.1 Combined test beam

A full slice of the ATLAS detector (inner tracker, calorimetry, muon spectrometer) has been tested in 2004 on CERN's H8 beam line. The simulation infrastructure is successfully managing all the different aspects of such a complex experimental setup. Simulation of the subdetectors must be possible in both combined and stand-alone modes, while beams with different particles, energies and profiles must be simulated; extra detectors, such as scintillators, must be properly handled, together with some extra dead material used for material studies.

All the above mentioned simulation parameters are indeed fully accessible from the python interface for the advanced user. Moreover, a database is provided associating the different configurations to a run number: a user only needs to specify in the jobOptions the run number, and the simulation infrastructure will automatically setup the corresponding set of parameters. Figure 3 shows a view of the CTB experimental setup.

2.2 Cosmic simulation

G4ATLAS provides full support for cosmic simulation, allowing the user to choose a specific setup consisting of the ATLAS detector, the ATLAS cavern, the rock overburden

Figure 4: View of the ATLAS cavern as described in the simulation. The Muon Chambers are shown as well.

and the surface buildings. The geometrical description of the different elements is provided by GeoModel, and primary cosmic generation is done via a specific generator (Cosmic-Generator) providing single muons at surface level. A scoring layer surrounds the ATLAS detector: particles crossing it will be saved to an external file, which can then be used to start a new simulation without having to propagate particles through the rock overburden. Figure 4 shows an example of geometrical setup used in the cosmics simulation, consisting of the Muon Chambers and the ATLAS cavern. The rock overburden, surface buildings and main shafts are simulated as well, although not shown in the picture.

Several detectors have expressed their concerns on the possibility that their existing detector-specific simulation software may not be able to properly handle hits with non standard timing (i.e., produced by tracks not originating at $t=0$ in the interaction point) such as the ones created in cosmic events. A request to have a fixed time reference common to all the subdetectors has thus been presented. The implemented solution consists of two virtual detectors, one placed in the inner surface of the LAr calorimeter and one on the outer surface of the Tile calorimeter. These scoring layers record positions, momenta and times of the primary muon and store them in the output file, together with the hits produced by the subdetectors. This information will thus be available for further analysis. With the addition of these two layers to the one surrounding the muon spectrometer, all the subdetectors are indeed surrounded by control layers recording the absolute time of muons: this provides all the information needed to solve whatever timing issue each individual subdetector may find in its simulation code.

3 Simulation validation

The validation efforts, which are done in parallel with the development of the simulation software, aim to spot any non optimal performance, internal inconsistency or inaccurate description of detectors or physical processes.

The simulation software is validated using three types of tests:

- Computing performance: the performance is measured in terms of CPU time per event and memory usage at run time
- Comparison with real data: the results of the simulation are compared with real data from test beams.
- Physics performance: reconstruction of simulated full physical events is performed and the results analyzed.

3.1 Computing performance

As simulation software is being used for massive productions on the GRID, its performance in terms of CPU time per event and memory usage at run time needs to be constantly monitored. Both these quantities are indeed measured at each nightly build of the ATLAS offline software by a set of dedicated jobs using different single particle samples. More detailed tests, including full physical events, are performed at each Athena release.

3.1.1 Memory usage

Memory usage at run time is measured at different steps of any job startup (Athena initialization, GeoModel initialization and G4ATLAS startup), as well as during the event processing. Moreover, at each nightly build, plots are automatically produced showing the memory usage as a function of the number of processed events, in order to easily spot any major software problem such as memory leaks. Figure 5 shows the results of such a measurement, obtained by using single muons in Athena release 10.5.0. The first step, occurring at event 49 is due to a known Geant4 problem, which causes the memory allocated in case of events with many tracks not to be released until the run processing ends. As shown the memory usage is constant (apart from the deallocation problem) during the run processing.

Figure 5: Memory usage at run time as a function of the number of processed events. The plot refers to single muons with $P_t = 5\text{GeV}$.

Figure 6 shows the memory usage at run time for the different steps of the G4ATLAS initialization, as measured in different Athena releases. The data plotted in the figure is also reported in Table 2. Release 10.4.0 was the first one in which the GeoModel version of the geometry for the LAr calorimeter was used. This caused a drop in the memory requirement for the LAr geometry, as expected. The G4Init entry represents the memory required by Geant4's internal initialization (geometry voxelization, cross section tables calculation), which was not monitored before 10.4.0. The value measured in 10.4.0 is therefore assigned also to older releases to allow easy comparison of the total Geant4 memory requirement. The changes in the memory required by Geant4 to load the physics list are related to changes in the underlying Geant4 version, rather than to changes of the ATLAS physics lists themselves.

In addition to the memory requirements of G4ATLAS itself, two other parts of a

Figure 6: Memory usage at run time for different steps of G4ATLAS initialization.

Release	Memory usage (MBytes)							
	ID	LAr	Tile	Muon	Magnetic	Physics	G4Init	G4 total
	geo+SD	geo+SD	geo+SD	geo+SD	field	list		
9.0.4	7.5	94.8	0.1	9.9	40.0	2.0	130.83	285.13
10.0.1	0.4	94.1	1.1	10.7	40.0	1.0	130.83	278.13
10.0.3	0.4	96.2	0.1	9.4	40.0	1.0	130.83	277.93
10.0.4	0.4	95.1	1.1	9.0	40.0	1.0	130.83	277.43
10.3.0	0.4	96.1	1.1	9.5	40.0	2.5	130.83	280.43
10.4.0	1.3	43.0	0.1	11.0	39.0	2.8	130.83	228.03
10.5.0	1.4	42.2	0.1	11.1	40.0	1.5	130.88	227.18
11.0.0	1.4	42.3	0.1	11.1	39.0	1.5	133.17	228.68
11.0.1	1.4	49.2	0.1	10.1	39.0	1.5	125.88	227.23

Table 2: Memory usage at run time for different steps of G4ATLAS initialization, as measured in different Athena releases.

Figure 7: Memory usage at run time for a generic simulation job.

generic simulation job are monitored, namely: Athena startup and GeoModel. Figure 7 shows the memory required by these two additional components, together with the total G4ATLAS memory. The sum of the three entries, giving the total memory usage at the beginning of the first event, is plotted as well.

3.1.2 CPU time measurements

The average CPU time per event is measured for both simulation of single particle samples and full physics events. A summary of the results obtained with single particles in different release is shown in Figure 8. As shown, the timing performance remained almost constant in the releases following the Rome production. The only exception is for release 11.0.0, where Geant4 was configured in order to ensure a more precise tracking of the particles in the magnetic field (see 3.1.5 for more details). This tuning, which was needed mainly by the Inner Detector, was indeed applied to the whole ATLAS detector. Being more precise, the tracking became also more sensitive to inconsistencies in the geometry descriptions of the subdetectors. In particular the calorimeters and the muon spectrometer were causing several navigation problems, resulting in an increase of the processing time per event. This is particularly evident in Figure 8(a), but present in all the single particle samples. In release 11.0.1, the precision of the tracking was increased only in the Inner Detector. This led to a return of the processing time per event to values consistent with the ones observed before release 11.0.0.

Starting from release 11.0.0, the decision was taken to rely, for the timing measurements, only on automated tests to be performed on each nightly build using the Run-

	CPU time per event (kSI2K)			
	9.0.4 (1000 events)	10.4.0 (100 events)	11.0.1 (1000 events)	11.0.2 (200 events)
Channel:				
B4 (jets)	707.73	1087.58	874.26	720.42
H(130) $\rightarrow$ 4l	829.81	1256.29	1028.92	901.19
minimum bias	275.01	498.50	379.87	263.78

Table 3: CPU time per event for full physics events, as measured in different releases.

TimeTester package. Given the limited computing resources, and the need to have all the test jobs completed in a short time, the most time consuming samples were dropped.

In addition to the nightly single particle tests, automated extensive tests are also performed after each new Athena release, with the goal of validating the release before the distribution kits are built. Since the time for a job to complete can in this case be extended to a few days, physics samples are included in the tests, although the limitation on computing resources forces the number of samples to be simulated to be small (4 channels). Table 3 shows the timing results for three types of physics events as measured in three different Athena releases: 9.0.4 is the release used for the Rome production; 10.4.0 was tested with limited statistics and 11.0.1 was the first to implement the automated tests of physics samples. The increase in time observed in 10.4.0 and 11.0.1 with respect to 9.0.4 is mostly related to the lack of direction filtering on the primary particles in releases starting from 10.4.0. Direction filtering is activated by default starting from release 11.0.2: preliminary timing measurements performed on this release are also reported in Table 3, and show good agreement with the results obtained using release 9.0.4, although a slight increase of about 5% is observed, due in part to the more precise tracking in magnetic field requested by the Inner Detector.

A summary of the tests done in the past years can be found in [5].

3.1.3 Reproducibility

Reproducibility has been an open issue for G4ATLAS during the latest releases, as reported in [5]. The first event of each run appeared to be exactly reproducible, while some instability was introduced during the run, which made it impossible to obtain identical results starting from identical job configurations.

A careful investigation showed that the problem was in the way events are read in from POOL: for a given vertex, its particles were not always read in the same order. The difference in the ordering of the particles leads indeed to slightly different simulation results, since particles processed in different order will be simulated using different random numbers. With release 11.0.0, a fix for this problem has been introduced on the simulation side, consisting in a reordering of the particles in each vertex before passing them to Geant4 for processing.

(a) Single muons

(b) Single pions

(c) Single electrons

Figure 8: Timing results for single particles.

sample	number of events	nightly run
single e, $P_t=5$ GeV, $ \eta < 6.0$	300	yes
single e, $P_t=50$ GeV, $ \eta < 2.5$	300	yes
single pi, $P_t=5$ GeV, $ \eta < 6.0$	300	yes
single pi, $P_t=50$ GeV, $ \eta < 3.2$	300	yes
single mu, $P_t=5$ GeV, $ \eta < 6.0$	300	yes
single mu, $P_t=50$ GeV, $ \eta < 3.0$	300	yes
B4 (jets)	250	no
H(130) $\rightarrow$ 4l	250	no
minimum bias	250	no
SUSY	250	no

Table 4: Summary of automated tests.

3.1.4 Automated tests

As already mentioned, release testing is now done in a completely automated way. The list of jobs which are run at every Athena release is shown in Table 4, where the last column tells whether the job is run during the nightly tests as well. For each job, the information concerning the memory usage and the CPU time per event is saved and archived, together with summary plots such as the one shown in Figure 5.

3.1.5 Performance optimization

LAr range cut. One of the handles that can be used to optimize a G4 simulation is the range cut on secondary particles. Only secondaries with a range greater than the cut are actually tracked by G4, while all the others are killed as soon as they are generated (their energy is of course properly accounted for). The default cut for the ATLAS simulation

Figure 9: Memory usage at run time for a generic simulation job.

Material	Particle	Range cut	Energy cut
Concrete	e^+/e^-	15cm	80MeV
	γ	15cm	123keV
Air	e^+/e^-	2m	600keV
	γ	2m	4keV
Rock	e^+/e^-	1km	-
	γ	1km	-

Table 5: Range cuts optimized for cosmic simulation

is 1mm. Different detectors can set different cuts, according to their needs. The LAr calorimeter, in particular, has chosen to set an extremely low range cut, lowering the default up to $30\mu\text{m}$. This has of course a serious impact on timing performance. In order to better understand the variations on the timing performance when changing the LAr range cut, a dedicated test was performed. Single electrons with $P_t=50\text{GeV}$ have been fired in a narrow η range (0.35-0.45), and only the LAr detector has been simulated. Time measurements have then been done for 14 different values of the range cut. The results are shown in Figure 9. As expected, the time per event increases rapidly when decreasing the range cut. In particular, going from $30\mu\text{m}$ to $60\mu\text{m}$ for example, would produce a decrease of about 10% in the CPU time per event.

Cosmic ray simulation. When performing the simulation of cosmic rays for ATLAS, a great part of the CPU time per event is used to propagate the primary muon through the rock overburden. Being the default range cuts set to 1mm, a great number of secondaries are produced and propagated into the rock. This can be avoided by setting the range cuts in the rock to some high value (such as 1km), so that no secondaries will be generated at all in the rock. In addition to rock, the overburden can be schematically described as composed by two other regions: the concrete of the cavern walls, and the air in the shafts and in the cavern. In a cosmic event, the average hit multiplicity is in general greater than the one that can be expected from the single primary muon. This is due to showers started by the primary muon in its path toward ATLAS. In order to find a proper value for the cuts in the concrete and air regions, several values were tested, and the set has been chosen which did not modify the average hit multiplicity, while still allowing for some time optimization. The resulting set of cuts is shown in Table 5. When applying these cuts, the average time per event for the simulation of the bare overburden (i.e., without the ATLAS detector) is 0.16 s, which means a factor 20 lower than the one measured with the standard 1mm cuts.

Tracking in magnetic field. When a simulated track enters a volume with a magnetic field, the Geant4 kernel steers the following tasks:

- the trajectory the particle is going to follow is calculated (with analytical or numerical methods)

	Default	Set A	Set B	Set C	Set D
DeltaIntersect	1 μm	0.1	0.01 μm	0.01 μm	0.01 μm
DeltaOneStep	10 μm	1 μm	0.1 μm	1 μm	10 μm

Table 6: Geant4 settings for tracking in magnetic field

	Time per event (kSI2K)				
	Default	Set A	Set B	Set C	Set D
ID only	0.138	0.157	0.238	0.232	0.178
Calo only	1.038	1.061	0.987	1.037	1.089
Muon only	0.461	0.592	0.520	0.465	0.612
All	2.099	2.403	2.147	2.168	2.026

Table 7: Timing performance for different tracking parameters

- linear steps are done along the theoretical curve
- these steps are used for simulation (i.e. physical processes and hit production)

The effect of the approximation of the real curve with linear steps is a systematic misplacement of the hits with respect to the “true” position, and an overall underestimation of the particle momentum. These biases are however known, and easily reducible to negligible values by carefully tuning some specific G4 parameters. On the other hand, a careful estimation of the impact of these parameters on timing performance is necessary. A sample of single muons with 100 GeV energy has thus been used for detailed performance tests, whose results are reported in Tables 6 and 7. The DeltaIntersect parameter controls the accuracy of the intersection of the linear steps with volume boundaries, with respect to the intersection of the theoretical curve. DeltaOneStep sets instead the overall position error which is allowed in one step, and is not expected indeed to introduce any bias. Several combinations of these two cuts were tested, and the set was chosen (Set B, in the tables) which produced good physics results while not degrading too much the timing performances.

3.2 Comparison with real data

The 2004 combined test beam (CTB), already described in 2.1, has been an important benchmark to evaluate the reliability of the simulation, and to compare real data with simulation. For all the subdetectors, analysis has shown very good agreement between the simulated response and the one measured at the test beam. We show in the following some measurements, representative for the different subdetectors.

The CTB setup included a Cherenkov counter, which was used, together with the calorimeters, to identify electrons and pions. This allowed for a comprehensive study of the particle identification capabilities of the ATLAS TRT. Figure 10 shows the pion efficiency as a function of the electron efficiency for a 2GeV beam. Identification is done

in the TRT by measuring the fraction of high level hits per track, which is in general higher for electrons than for pions. The comparison of real data with simulation shows a good agreement.

Concerning calorimeter performance, Figure 11 shows the longitudinal shower profile for a 100 GeV electron beam in both data and simulation. Figure 12 shows instead the combined response of electromagnetic and hadronic calorimeter under a muon beam. Good agreement is found between data and simulation, which remarkably is not degraded in the tails of the distributions.

Sagitta measurements in the muon spectrometer allow to make detailed comparisons between data and simulation. The measured sagitta width has two main contributions: the intrinsic detector resolution (independent on the muon momentum) and the multiple scattering (function of the muon momentum). Figure 13 shows the sagitta resolution (standard deviation of the sagitta distribution) as a function of the incident muon momentum, for both data and simulation. As shown, the agreement is good at low momenta, thus confirming that the material description used in the simulation is correct, and that the multiple scattering process is well simulated. The disagreement at higher momenta (where the multiple scattering contribution becomes negligible) is due to the digitization code not reproducing perfectly the intrinsic detector resolution.

3.3 Physics performance

Complete reconstruction of full physics events simulated with G4ATLAS is an important part of the simulation validation process, since any anomaly in the reconstructed quantities allows to spot problems either on the simulation itself or in the geometrical description of the detector. The need of a finer tuning of the parameters controlling the accuracy of the tracking in a magnetic field (see 3.1.5), was indeed discovered studying

Figure 10: TRT pion rejection power for low energy particles. For more information, see [6].

Figure 11: Longitudinal shower profiles for LAr calorimeter.

Figure 12: Total energy deposit in the calorimeters for muons. For more details, see [8].

Figure 13: Sagitta resolution in the muon spectrometer, as a function of the muon energy. For more details see [9]

discrepancies between reconstructed and simulated momenta.

Figure 14 shows the results of a b-tagging study, where the light jet rejection is plotted versus the b-tagging efficiency. Results obtained with older Geant3 simulations are reported as well, showing good agreement with G4ATLAS results.

4 Conclusions

The simulation software of the ATLAS detector, based on the GEANT4 toolkit, is presently a robust and full featured application suite, integrated in the ATLAS offline software. The needed flexibility, configurability and ease-of-use are achieved through python scripts, which allow the users to have full control on the simulation job. The simulation suite is being successfully used for massive production since 2004, and its performance is continuously monitored using automated nightly tests, as well as more detailed measurements at each software release.

References

- [1] *Physics Generation for ATLAS Data Challenges* webpage, maintained by I. Hinchliffe, <http://www-theory.lbl.gov/~ianh/dc/dc2.html>.
- [2] The GEANT4 Collaboration (S. Agostinelli *et al.*), *GEANT4 — A Simulation Toolkit*, Nuclear Instruments and Methods in Physics Research, NIM **A** 506 (2003), 250–303.
- [3] D. Barberis, G. Polesello, A. Rimoldi, *Strategy for the transition from Geant3 to Geant4 in ATLAS*, ATL-SOFT-2003-013.
- [4] The Athena framework:
<http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/architecture/General/Documentation/AthenaDeveloperGuide-8.0.0-draft.pdf>

Figure 14: b-tagging efficiency vs light jet rejection. Comparison with old Geant3 data shows good agreement. For more details see [10]

- [5] D. Costanzo, A. Dell'Acqua, M. Gallas, A. Nairz, N. Benekos, A. Rimoldi, J. Boudreau, V. Tsulaia, *Validation of the GEANT4-Based Full Simulation Program for the ATLAS Detector*, ATL-SOFT-PUB-2005-004.
- [6] Vu Anh Tuan, *ID material studies and update on electron identification*, presentation at the “ATLAS Software and Computing Workshop”, CERN, Sept. 26–30, 2005 agenda <http://agenda.cern.ch/fullAgenda.php?ida=a045109>.
- [7] L. Serin, *Electromagnetic calorimeter performances in beam tests with production modules*, presentation at the “ATLAS Physics Workshop”, Rome, June 6–11, 2005; agenda <http://agenda.cern.ch/fullAgenda.php?ida=a044738>
- [8] G. Avolio, *Combined Muon Performance with Calorimeter Information at H8*, presentation at the “ATLAS Physics Workshop”, Rome, June 6–11, 2005; agenda <http://agenda.cern.ch/fullAgenda.php?ida=a044738>
- [9] D. Rebuzzi, S. Rosati and G. Avolio, *Studies on H8 MDT Barrel Simulation*, presentation at the “Vth Workshop on H8/X5 Muon Test”, CERN, Dec. 16–17, 2004; agenda <http://agenda.cern.ch/fullAgenda.php?ida=a044309>.
- [10] L. Vacavant, *b-tagging performances with Rome samples*, presentation at the “ATLAS Physics Workshop”, Rome, June 6–11, 2005; agenda <http://agenda.cern.ch/fullAgenda.php?ida=a044738>