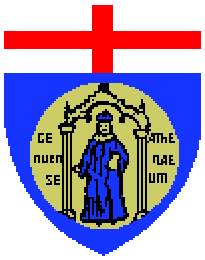
UNIVERSITÀ DEGLI STUDI DI GENOVA



DOTTORATO DI RICERCA IN FISICA
XIV CICLO

FABRIZIO FERRO

Reconstruction of primary and decay vertices in the ATLAS experiment at LHC and study of the $A \rightarrow \tau\,\tau$ channel

Tesi di dottorato

Tutore: Prof.ssa Bianca Osculati

Tutore esterno: Dott.ssa Donatella Cavalli

Coordinatore: Prof. Mauro Giannini

Contents

Chapter 1

The LHC and the ATLAS experiment

This chapter provides an overview of the CERN LHC project and of the ATLAS experiment. The ATLAS subdetectors layout and parameters, even if frozen in their main features, are still undergoing optimization processes. We refer here to the apparatus as described in the *Technical Design Report* [1], which could be out of date in some details. The *Pixel Detector* description is the most up to date, since the *Vertexing* tool (described in chapter 3) depends mainly on its performances.

1.1 The LHC accelerator

The LHC (*Large Hadron Collider*) is a new generation proton-proton collider currently under construction at CERN. It will be installed in the same 27 km long tunnel which hosted the LEP accelerator. The LEP machine and the four experimental apparatus exploiting this accelerator have been completely removed. At the time being, civil engineering works have not yet been completely finished and the construction of the challenging superconducting magnets and of other accelerator components is still in progress. The start of the LHC activities is scheduled for the year 2007.

Even if the location of the LHC will be the same as LEP, the characteristics of the two accelerators are considerably different. LEP accelerated electron and positron beams in opposite directions inside the same beam pipe; while LHC will accelerate two protons beams running in opposite directions inside two parallel beam pipes (see fig. 1.1). The highest energy reached by LEP has been around 105 GeV per beam, while LHC will operate at 7 TeV: the energy available in the collision between the proton constituents (quarks and gluons) will be in the TeV range.

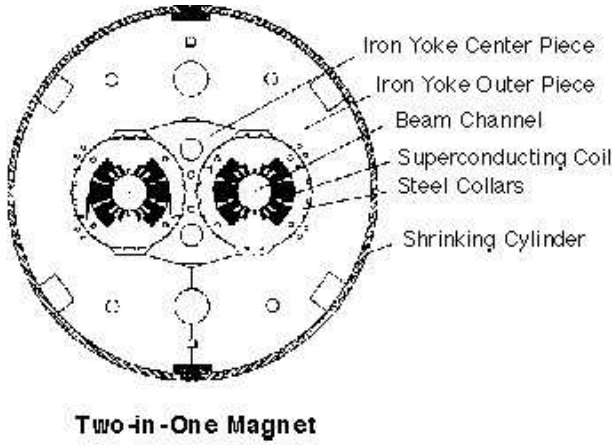


Figure 1.1: *LHC section.*

The *synchrotron radiation* emitted by the particles accelerated in the ring is reduced in the LHC by a factor $(m_e/m_p)^4$ w.r.t. an electron accelerator of the same energy. Nevertheless, the *synchrotron radiation* have not to be undervalued: the cryogenic apparatus must absorb the emitted power which amounts to about 3.7 kW and the photoelectrons produced can create beam instability once accelerated by the electric field.

The LHC will operate at low and high luminosity[1] (respectively 10^{33} and 10^{34} $cm^{-2}s^{-1}$); at high luminosity, the LHC will be filled with 2835 bunches of 10^{11} protons each, yielding a current of about 0.53 A.

The main LHC features are described in tab. 1.1.

The LHC magnets must be able to produce fields of 8.36 Tesla, in order to bend 7 TeV protons. This can be achieved using superconducting materials, such as copper-clad niobium-titanium, for the cables in the magnets coils. Since superconductivity needs very low temperatures, the cryogenic LHC apparatus has to keep the coils temperature at 1.9 K. It will be achieved using superfluid Helium,

[1]The luminosity $\mathcal{L}$ is defined by the following relation:

$$N = \mathcal{L}\sigma$$

with N and σ respectively the number of event expected and the cross-section of a particular process. The luminosity is strictly correlated to the properties of the accelerator and can be also expressed as

$$\mathcal{L} = f\frac{n_1 n_2}{4\pi\sigma_x\sigma_y}$$

where f is the frequency at which the collisions occur, n_1 and n_2 are the number of particles inside the colliding bunches and σ_x and σ_y are the standard deviations of the Gaussian transverse beam profile.

LHC

Max. beam energy	7 TeV
Luminosity	10^{34} $cm^{-2}s^{-1}$
Time between collisions	25 ns
Bunch length	7.5 cm
Beam radius	15 μm
Free space at interaction point	38 m
Luminosity lifetime	10 h
Filling time	6 min
Acceleration period	1200 s
Injection energy	450 GeV
Peak magnetic field	8.3 T
RF frequency	400.8 MHz
Particles per bunch	10^{11}
Number of bunches	2835

Table 1.1: *LHC main properties.*

which an appropriate pipeline system will distribute to all the 5000 magnets in the accelerator.

The protons are injected into the LHC after having been accelerated up to 450 GeV energy by the CERN injection system (see fig. 1.2). It consists of a chain of accelerators: *LINAC*, *Booster*, *PS* and *SPS*.

The LHC accelerator will also be able to accelerate heavy ions, such as Pb ions, at energies of about 2.76 TeV. This will allow the dedicated experiment to acquire data and to study the heavy ions physics processes.

The proton-proton (ion-ion) collisions will take place in four different interaction points at which five experiments will deploy their detectors to acquire physical data. These experiments are:

- ALICE, which will study heavy ions physics;

- ATLAS and CMS, which will mainly look for Higgs and supersymmetric particles;

- LHCb, which will study b-physics;

- TOTEM, which will lay inside CMS and will measure proton-proton total and elastic cross-section.

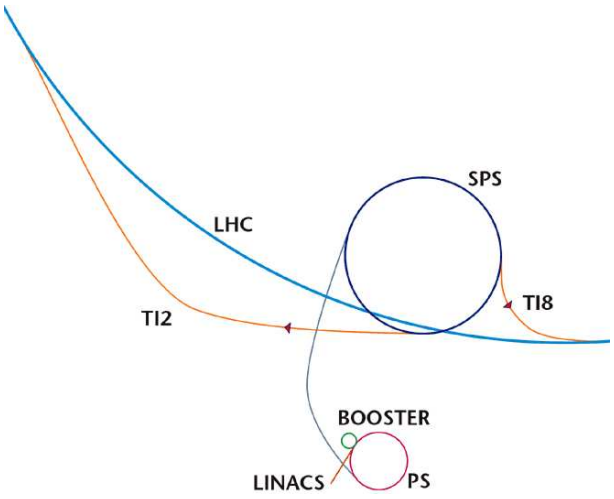


Figure 1.2: *LHC injection system.*

1.2 The ATLAS experiment

The ATLAS experiment is one of the five experiments which will take data at the LHC accelerator. It is managed by a world-wide collaboration which includes over 2000 physicists from many different institutions.

The purpose of this experiment is to collect and analyze the data provided by the LHC proton-proton collisions. The physical aim is to investigate an energy range never reached before, to find new particles and to test physical theories which still need to be proved (or discharged).

In order to achieve this goal, a very high technology detector is under construction in several institutes and laboratories all over the world. The detector has been designed following a few important basic criteria:

- very good electro-magnetic calorimetry for electron and photon identification;

- full-coverage hadronic calorimetry to provide good jet identification and missing transverse energy (E_T^{miss})measurement;

- high precision muon momentum measurement;

- very efficient tracking;

- large acceptance in pseudorapidity (defined as $\eta = -log(tan(\theta/2))$, with θ angle from the beam axis) and full-coverage in the azimuthal angle ϕ;

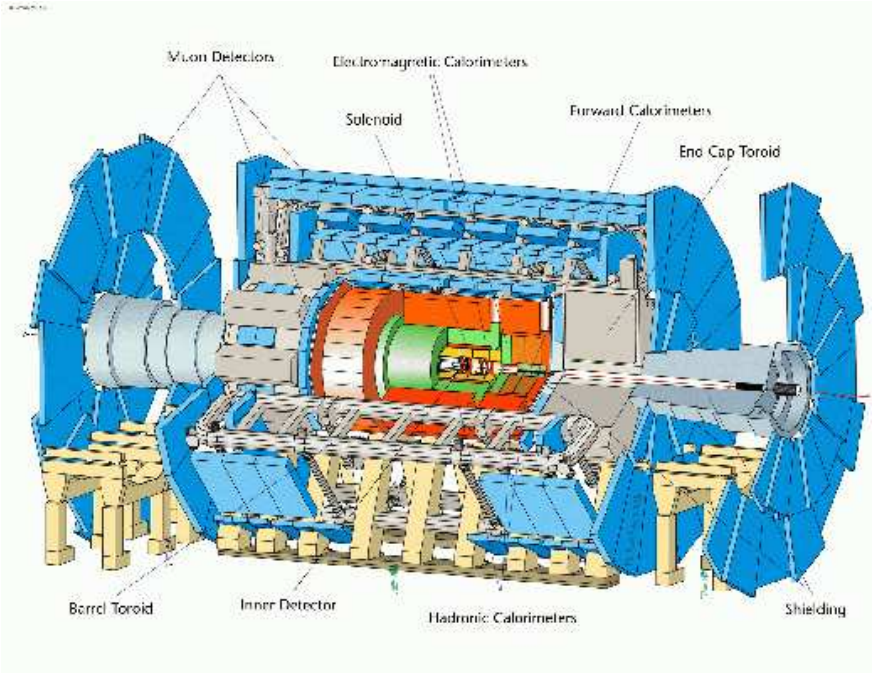


Figure 1.3: *The ATLAS detector (1).*

- triggering and measurement of particles at different p_T thresholds ($p_T = \sqrt{p_x^2 + p_y^2}$).

A detector picture can be seen in fig. 1.3. It is divided into three main regions: the central barrel and the two end-caps. In the barrel region, following the inside-out direction, one can find: the *Inner Detector*, the *Electro-Magnetic Calorimeter*, the *Hadronic Calorimeter* and the *Muon Spectrometer*.

1.2.1 Magnets system

As can be seen in fig. 1.4, the ATLAS superconducting magnet system is made of

- a *Central Solenoid* (CS) which surrounds the tracking elements to provide the necessary magnetic field (parallel to the beam axis);

- a system of three air-core toroids, two in the end-caps (ECT) and one in the barrel (BT), which generate the magnetic field for the *Muon Spectrometer*.

The CS is powered by a 8 kA power supply and provide a central field of 2 T with a peak magnetic field of 2.6 T.

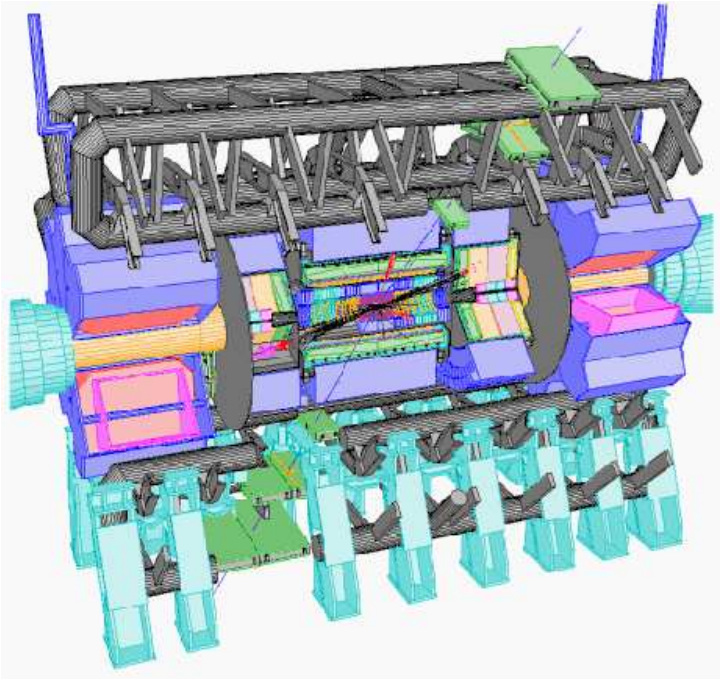


Figure 1.4: *The ATLAS detector (2).*

The peak magnetic field in the BT and ECT are 3.9 and 4.1 respectively. Both the BT and the ECT are made of eight coils cooled by a forced flow of helium at 4.5 K. They are powered by a 21 kA power supply and are equipped with a control systems for fast and slow energy dumps.

1.2.2 Inner Detector

The Inner Detector (ID) is the innermost part of ATLAS (see the layout in fig. 1.5). Its goal is to provide good track measurements, to optimize the momentum and vertex resolution. For this reason, fine-granularity detectors are needed, particularly in the part closer to the interaction region. The ID is made of three concentric subdetectors. Starting from the innermost, they are:

- the Pixel detector: made of 10^8 pixel sensors; its aim is to provide very high granularity to disentangle the tracks near the collision point, to allow impact parameter measurements and primary and secondary vertex position finding;

- the Semiconductor tracker: made of double layers silicon strips, which will measure precision points at larger radii;

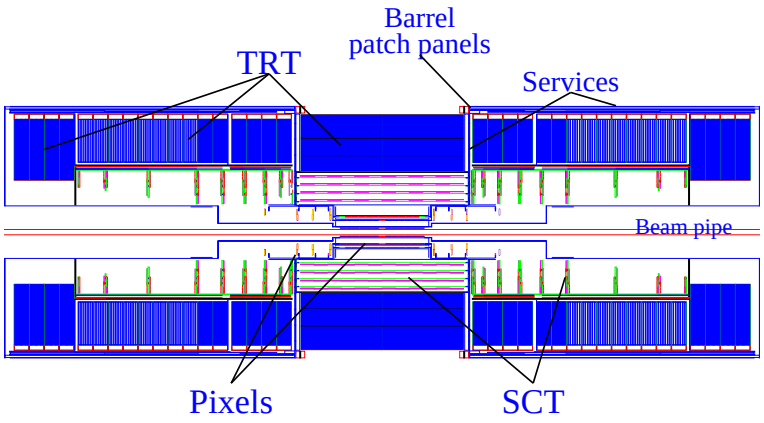


Figure 1.5: *The Inner Detector (TDR [1] geometry).*

- the Transition Radiation Tracker: made of a large number of straw detectors; it will provide continuous tracking at large radii and will enhance significantly the electron identification capability of the Inner Detector.

Pixel detector

The Pixel detector provides a high-granularity, high-precision set of measurements close to the interaction point. The complete system consists of three barrel planes, at radii of 5.05 cm, 8.85 cm and 12.25 cm, and of three disks at each side, at 49.5 cm, 58 cm and 65 cm. [2] The detector contains about 1700 modules and 28000 readout chips which yield about $8.5 \cdot 10^7$ readout channels. The 50 x 400 μm^2 pixel sensors are arranged in arrays of 320x144 pixels, read by 16 R/O chips *bump-bonded* directly on them. The data acquisition is managed by a MCC (Module Control Circuit) chip per module, which acquires data, builds the event and communicate with the external readout units. Each chip must be radiation hard to withstand over 300 kGy of ionizing radiation (ten years of operation). The detector performances have been measured with dedicated test-beam [2]: the spatial resolution is about 13 μm in the 50 μm coordinate and about 65 μm in the orthogonal one.

Semiconductor tracker

The SCT system provides eight precision measurements per track, contributing to the measurement of momentum, impact parameter and vertex position. The barrel part is made of eight double layers of silicon microstrips detectors at 40

[2]Because of economical issues, the detector will be initially provided with only 2 barrel layers and 2 disks at each side.

mrad stereoangle, with 768 readout strips of 80 μm pitch. They are mounted on four complete barrels at radii of 30, 37.3, 44.7 and 52 cm. The end-caps modules are mounted on nine wheels per side, providing an η coverage up to $|\eta| = 2.5$. The spatial resolution is 20 μm in $R\phi$ and 550 μm in z.

Transition Radiation Tracker

The TRT is made of about 420000 straw detectors 144 cm long and with a diameter of 4 mm. The detector provides tracking of charged particles and a very good electron identification capability, detecting transition radiation photons created in a radiator between the straws. These are filled with a gas mixture of 70% Xe, 20% CO_2 and 10% CF_4. The barrel section is made of individual modules covering the radial range between 56 and 107 cm, while the two end-caps consist each of 18 wheels.
About 35 points per track are measured, with a resolution of 150 μm.

1.2.3 Calorimeters

The ATLAS calorimetry (see fig. 1.6) consists of:

- the Electromagnetic calorimeter EM ($|\eta| < 3.2$);
- the Hadronic barrel calorimeter HCAL ($|\eta| < 1.7$);
- the Hadronic end-caps calorimeters HECs ($1.5 < |\eta| < 3.2$);
- the Forward calorimeters FCALs ($3.1 < |\eta| < 4.9$).

Electromagnetic calorimeter

The EM calorimeter is a lead/liquid-argon (LAr) detector with accordion geometry. Radially it extends from 1.5 m to 1.98 m. It is divided into a barrel part ($|\eta| < 1.475$) and two end-caps ($1.375 < |\eta| < 3.2$). The barrel calorimeter consists of two identical half-barrel divided by a 6 mm gap, while each end-cap calorimeter is divided into two coaxial wheels covering $1.375 < |\eta| < 2.5$ and $2.5 < |\eta| < 3.2$. Over the region devoted to precision physics ($|\eta| < 2.5$) the EM is segmented into three longitudinal sections: the strip section, which is equipped with narrow strips and which acts as a *preshower* detector, enhancing particle identification, the middle section, which is transversally segmented into towers of size $\Delta\eta$x$\Delta\phi = 0.025$x0.025 and the back section, which has a granularity of 0.05 in η and a thickness varying between 2 and 12 X_0 (*radiation length*).

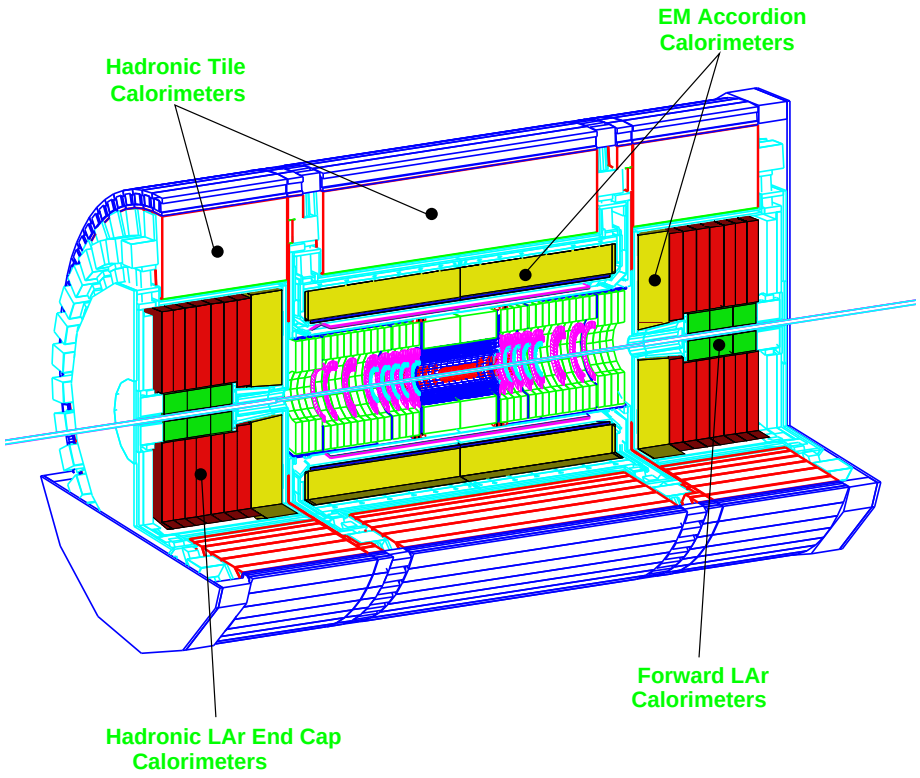


Figure 1.6: *The ATLAS calorimeters.*

Tile hadronic calorimeter

The large hadronic barrel calorimeter is a sampling calorimeter which uses iron as absorber and 3 mm thick scintillating plates (tiles) as active material. Two sides of the scintillating tiles are read out by wavelength shifting fibers into two separate photomultipliers. The tile calorimeter is composed of one barrel and two extended barrels. Radially it extends from 2.28 m to 4.25 m. It is longitudinally segmented into three layers, while azimuthally the barrel and extended barrels are divided into 64 modules.

Liquid-argon hadronic end-cap calorimeter

The HECs consist of two independent wheels of radius 2.73 m. Each wheel is made of 32 identical modules. The 8.5 mm gap between consecutive copper plates is equipped with three parallel electrodes, splitting the gap into four drift spaces.

Liquid-argon forward calorimeter

The FCAL is a very challenging detector, since it has to cope with a very high level of radiation. It is divided into three parts, one made of copper and two made of tungsten. The calorimeter consists of a metal matrix with regularly spaced longitudinal channels filled with concentrical rods and tubes; the rods are at positive high voltage, while the matrix and tubes are grounded.

1.2.4 Muon spectrometer

The Muon spectrometer is the outermost ATLAS subdetector (the layout can be seen in fig. 1.7). Its aim is to reconstruct muon tracks and measure their momentum, exploiting the magnetic field produced by the air-core toroid magnets. The spectrometer is made of separate trigger and high-precision tracking chambers. In the barrel region, the chambers are arranged in three cylindrical layers (*stations*) at radii of about 5, 7.5 and 10 m, while in the end-caps regions the chambers are installed vertically. Over most of the η-range, a precision measurement of the tracks position is provided by Monitored Drift Tubes (MDT). At large pseudorapidity Cathode Strip Chambers are used, since higher granularity is necessary to withstand the demanding rate and background conditions.

The trigger system covers the pseudorapidity range $|\eta| < 2.4$; Resistive Plate Chambers (RPC) are used in the barrel and Thin Gap Chambers (TGC) in the end-caps regions.

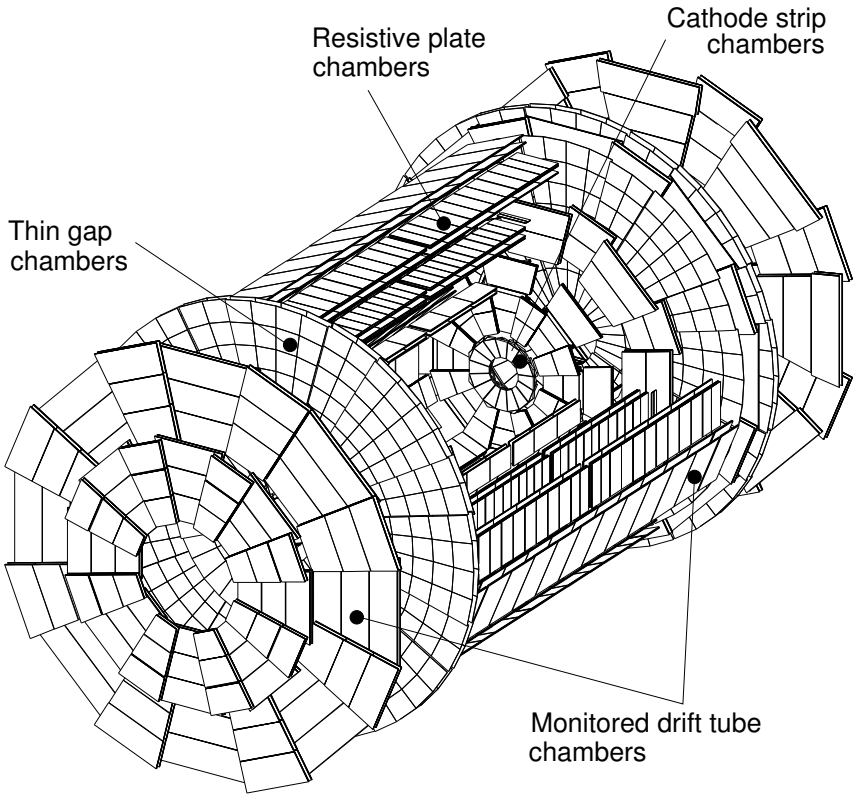


Figure 1.7: *The Muon spectrometer.*

Monitored drift-tube chambers

The basic detection elements of the MDT chambers are aluminium tubes of 30 mm diameter and 400 μm wall thickness, with a 50 μm diameter central W-Re wire. The tubes are operated with a gas mixture of 93% Ar and 7% CO_2 at 3 bar absolute pressure. The single wire resolution is about 80 μm.

Cathode strip chambers

The CSCs are multiwire proportional chambers with cathode strips readout and with a symmetric cell in which the anode-cathode spacing is equal to the anode wire pitch. Good spatial resolution is achieved by segmentation of the readout cathode and by charge interpolation between neighbouring strips. The cathode strips are orthogonal to the anode wire. The anode wire pitch is 2.54 mm and the cathode readout pitch is 5.08 mm. CSCs provide small electron drift times (30 ns) and good time resolution (7 ns). The gas used is a mixture of 30% Ar, 50% CO_2 and 20% CF_4.

Resistive plate chambers

The RPC is a gaseous detector providing a typical space-time resolution of 1 cm x 1 ns with digital readout. The basic RPC unit is a narrow gas gap formed by two parallel resistive bakelite plates, separated by insulating spacers. The primary ionization electrons are multiplied into avalanches by a high, uniform electric field of typically 4.5 kV/mm. The gas used is tetrafluorene ($C_2H_2F_4$) with a small admixture of SF_6. The signal is read out via capacitive coupling by metal strips on both sides of the detector.

Thin gap chambers

The TGCs are similar in design to multiwire proportional chambers but with the anode wire pitch larger than the cathode-anode distance. Signals from the anode wires provide the trigger information together with read out strips arranged orthogonal to the wires. The gas is a highly quenching mixture of 55% CO_2 and 45% n-C_5H_{12}. The cathode-cathode distance is 2.8 mm, the wire pitch is 1.8 mm and the wire diameter is 50 μm.

1.3 Trigger and Data Acquisition

The ATLAS trigger and data-acquisition system is based on three levels of event selection: LVL1, LVL2 and Event Filter (see fig 1.8).

The system has to withstand the very high LHC interaction rate and must be able to handle very large amounts of data. The bunch-crossing rate is $R_{bc} = 40$

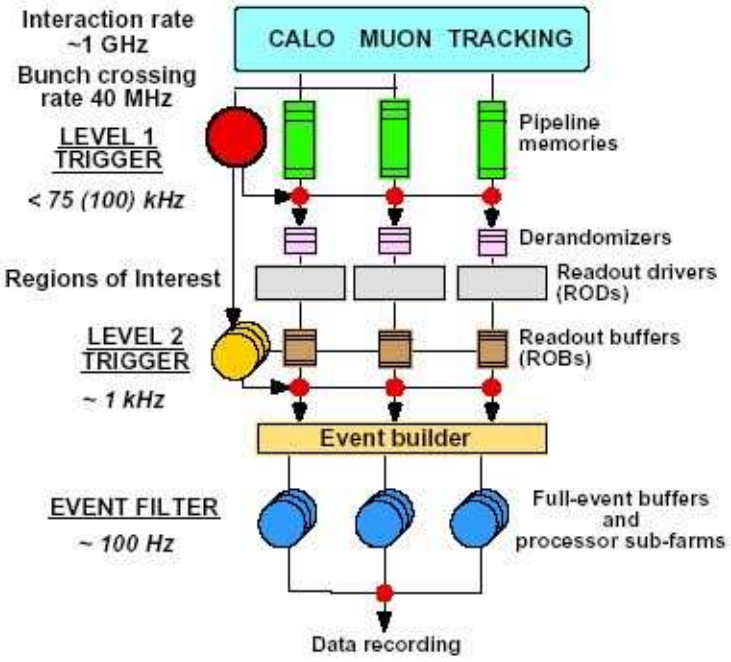


Figure 1.8: *The Trigger/DAQ system.*

MHz at high luminosity, the rate of the selected events must be reduced by a factor 10^7 ($R_{sel} \simeq 100$ Hz) for permanent storage. The challenge is to achieve the desired rejection upon *minimum-bias* events, keeping, on the other hand, a very high efficiency for the rare new physics processes.

The LVL1 trigger makes a selection based on reduced granularity information from a subset of detectors. It looks for high-p_T muons using the RPCs and the TGCs and for electrons and photons, jets, τ-leptons and large missing energy, using the calorimeters. The time to form and distribute the LVL1 decision (*latency*) must be minimized, in order to avoid loss of data. These are temporary stored in pipeline memories which can contain the data taken by each channel for the time needed to make a LVL1 decision. The required *latency* is about 2.5 μs. The data are first staged in the front-end electronics and are then sent to the readout drivers RODs, and finally to the readout buffers ROBs.

LVL1 trigger selects the detector region (RoI) in which is located the most significant part of the event and send it to the LVL2 trigger. LVL2 analyses only the RoI, but with access to all data with full precision and granularity. The LVL1 rate is about 75 kHz, the LVL2 rate is about 1 kHz and its latency is 10 ms.

The data selected by LVL2 trigger are analysed by the event filter, which uses offline algorithms adapted to the online timescale. The final rate is expected to be about 100 Hz.

The algorithms to be used in the trigger selections are extremely important, since different selections can make possible (or impossible) a specific analysis. They must be chosen with great care. The discussion about these issues is still partially open.

1.4 Offline software

The ATLAS experiment will produce about 1 Pbytes of data per year. This enormous amount of data must be stored and analyzed by the more than 1000 physicists involved in the experiment and working in different institutes all over the world. This challenging scenario needs:

- a central storage facility, to be held at CERN, and a few storage geographically distributed centers;
- a well organized and solid software development and maintenance;
- a very high performance world-wide network.

CERN is doing very big efforts to provide an adequate storage system for the LHC experiments. A system which is mostly user-transparent and which is able to stage data both on disks and on permanent tapes (*CASTOR*) is already available at CERN and is being used by all the LHC collaborations. Several local storage system are being installed in the main countries involved in the project, in order to share the storage load and to speed up user access to data.

Differently to what happened for older and smaller experiments, the ATLAS software development has necessarily started a few years ago and needs the effort of lots of physicists and professional programmers. All the software developers have access to a *repository*, where the software is stored. The *repository* is equipped with a system for the version control (*CVS*) to keep record of all modifications made in the code, which is periodically *released* and tested by all users. The code is arranged in *packages*, which can be defined as self-consistent units of code and which are under the responsibility of a single developer or of a group. The object-oriented programming strategy has been chosen for the ATLAS software, and the *C++* language is, at the time being, the most widely used language. A framework (*Athena*) in which all the algorithms will be run has been created. It is being used as the official framework for the ATLAS software.

Part of the software is dedicated to the analysis of the data taken during the runs. The world-wide distributed computing power has to be organized and equipped with a system capable of handling very large amounts of data. The existing computing centers and the new ones under construction are going to be connected with high performances links. Moreover, a world-wide project, called

GRID, is under development. It will be able to create an automatic computing power managing system which, in the aim, will allow the users to access any connected facility transparently and efficiently.

Bibliography

[1] The ATLAS Collaboration
ATLAS Detector and Physics Performance Technical Design Report
CERN - LHCC 99/14-15

[2] I. Gorelov et al.
A measurement of Lorentz angle and spatial resolution of radiation hard silicon pixel sensors
ATLAS Note - to be published

Chapter 2

Theoretical introduction

This chapter is intended to be an overview of the Standard Model and of its minimal supersymmetric extension (MSSM), in order to provide the theoretical framework for the A Higgs boson search described in chapter 4. For a complete and detailed description of the Standard Model and of SUSY theories, a few textbooks and scientific papers are reported in the bibliographic section [1], [2], [8], [9], [10], [11].

2.1 The Standard Model

The Standard Model (SM) is the theory describing strong corrections to the electroweak interactions. In this model the forces acting on the fundamental particles (see table 2.1) are mediated by the exchange of spin-1 gauge bosons (see table 2.2).

leptons

e	μ	τ
ν_e	ν_μ	ν_τ

quarks

u	c	t
d	s	b

Table 2.1: *Fundamental particles.*

In the SM theory, the particles are represented as fields and their state equations are determined from the Lagrangian density distribution, which is invariant under local transformations of the following symmetry group:

$$SU(3)_{colour} \bigotimes SU(2)_{left} \bigotimes U(1)_{hypercharge}.$$

Force	*Boson*	*Mass* (GeV/c^2)
Electromagnetic	γ	0
Weak	Z^0, $W^{\pm}$	91.188, 80.45
Strong	g	0

Table 2.2: *Gauge bosons.*

A consequence of these local gauge symmetries is the existence of propagators which mediate the interactions between particle fields in different space-time points. These are the gauge fields, whose properties are shown in table 2.3.

	quantum numbers	*coupling constant*
gluons G^a_μ	$(\underline{8},\underline{1},0)$	g_s
intermediate weak bosons W^i_μ	$(\underline{1},\underline{3},0)$	g
abelian bosons B_μ	$(\underline{1},\underline{1},0)$	g'

Table 2.3: *Gauge fields. The quantum numbers shown in the second column correspond to the* $SU(3)\bigotimes SU(2)\bigotimes U(1)$ *representation.*

The physical states (i.e. mass and electric charge eigenstates) are linked to the gauge fields by the relations

$$W^{\pm} = \frac{1}{\sqrt{2}}(W_1 \mp iW_2)$$

$$W^3_\mu = cos\theta_w Z_\mu + sin\theta_w A_\mu$$
$$B_\mu = -sin\theta_w Z_\mu + cos\theta_w A_\mu$$

where A_μ represents the electro-magnetic field and θ_w is the so-called Weinberg angle. This parameter is also linked to the electric charge e:

$$gsin\theta_w = g'cos\theta_w = e.$$

The matter fields are fermions and they are organized in three generations. Since the theory is invariant under $SU(2)_{left}$ transformations, left-handed and right-handed fermions have different quantum numbers (see table 2.4).

An important feature of the SM is the way bosons and fermions masses are handled, since the bare mass terms cannot be introduced in the Lagrangian,

quarks			
$\begin{pmatrix} u_i \\ d_i \end{pmatrix}_L$	$\begin{pmatrix} c_i \\ s_i \end{pmatrix}_L$	$\begin{pmatrix} t_i \\ b_i \end{pmatrix}_L$	$(\underline{3}, \underline{2}, \frac{1}{3})$
u_{iR}	c_{iR}	t_{iR}	$(\underline{3}^*, \underline{1}, \frac{4}{3})$
d_{iR}	s_{iR}	b_{iR}	$(\underline{3}^*, \underline{1}, -\frac{2}{3})$

leptons			
$\begin{pmatrix} \nu_e \\ e \end{pmatrix}_L$	$\begin{pmatrix} \nu_\mu \\ \mu \end{pmatrix}_L$	$\begin{pmatrix} \nu_\tau \\ \tau \end{pmatrix}_L$	$(\underline{1}, \underline{2}, -1)$
e_R	μ_R	τ_R	$(\underline{1}, \underline{1}, -2)$

Table 2.4: *Particle fields split in the left-handed and right-handed parts. The quantum numbers shown in the last column correspond to the* $SU(3) \bigotimes SU(2) \bigotimes U(1)$ *representation.*

without spoiling the $SU(2) \bigotimes U(1)$ gauge invariance.
A mechanism to solve the problem was proposed by Higgs in 1964. He assumed the existence of a complex scalar boson field ϕ with a vacuum potential such as

$$V = \mu^2 \phi^\dagger \phi - \lambda (\phi^\dagger \phi)^2$$

which had infinite non-zero ground states. The choice of one of these states would lead to a spontaneous symmetry breaking, meaning that the $SU(2) \bigotimes U(1)$ symmetry is not directly broken in the Lagrangian but only by the particular choice of the ground state. The addition of this vacuum potential yields a massive scalar boson (the Higgs boson) and three massless so-called Goldstone bosons, which can be made disappear by mean of a proper choice of the gauge.
In this framework, the gauge bosons masses are

$$m_W = \frac{gv}{2}$$

$$m_Z = \frac{v}{2}\sqrt{g^2 + g'^2}$$

with v vacuum state expectation value of the Higgs field.

The fermionic masses, which remain free parameters of the theory, can be implemented into the Lagrangian by mean of the coupling with the Higgs boson. The leptonic masses are

$$m_l = \frac{g_l v}{\sqrt{2}}$$

with g_l free parameter that must be set properly to obtain the experimental mass value.

The quarks masses, instead, need to be expressed by mean of the matrices that describe the mixing between the quarks generations. The properties of

mixing in the quark sector are summarized in the so-called Cabibbo-Kobayashi-Maskawa matrix (*CKM-matrix*).

High precision tests performed at LEP and SLD experiments have shown an excellent agreement between measurements and SM predictions (excepted for the Higgs boson, which has not yet been discovered). The basic measurements performed to verify the SM are

- the hadronic pole cross section of Z exchange

$$\sigma^0_{had} = \frac{12\pi}{m_Z^2}\frac{\Gamma_{ee}\Gamma_{had}}{\Gamma_Z^2}$$

  where Γ_{ee} and Γ_{had} are the partial width of the Z for decays into electrons and hadrons

- the total decay rate

$$\Gamma_Z = \Gamma_{ee} + \Gamma_{\mu\mu} + \Gamma_{\tau\tau} + N_\nu\Gamma_\nu + \Gamma_{had}$$

- the partial decay rates

$$R_l = \frac{\Gamma_{had}}{\Gamma_{ll}}$$

$$R_{b,c} = \frac{\Gamma_{b\bar{b}}}{\Gamma_{ll}}, \frac{\Gamma_{c\bar{c}}}{\Gamma_{ll}}$$

$$\Gamma_{inv} = N_\nu\Gamma_\nu$$

- the forward-backward (A^f_{FB}) and left-right (A_{LR}) asymmetries

- the τ polarization (P^τ)

The experimental results are summarized in fig. 2.1.

Although the Higgs boson has not yet been observed, it has been possible to constrain its mass by direct and indirect measurements. LEP direct search has excluded Higgs mass values smaller than 114.4 GeV/c^2 at the 95% confidence level [4]. Moreover, indirect fits of the precision electro-weak data (e.g. radiative corrections to the W and *top* mass) indicate $m_H = 98^{+53}_{-36}$ (see fig. 2.2) [3].

Theoretical predictions are not stringent. Lower bounds can be set requiring vacuum stability, while an upper limit can be established requiring that the theory remains valid up to a cut-off energy scale Λ. As shown in fig. 2.3, only a small part of the theoretical allowed spectrum has been checked at the time being.

Although the SM has been experimentally verified at a high precision level, a few aspects remain unclear:

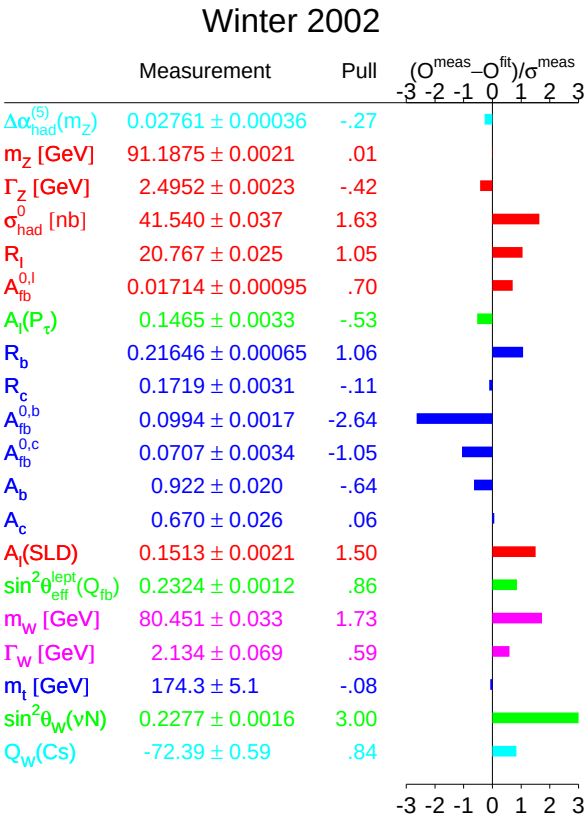


Figure 2.1: *Summary of the measured SM electro-weak parameters. The results of the pull between measured and fitted values are also shown.*

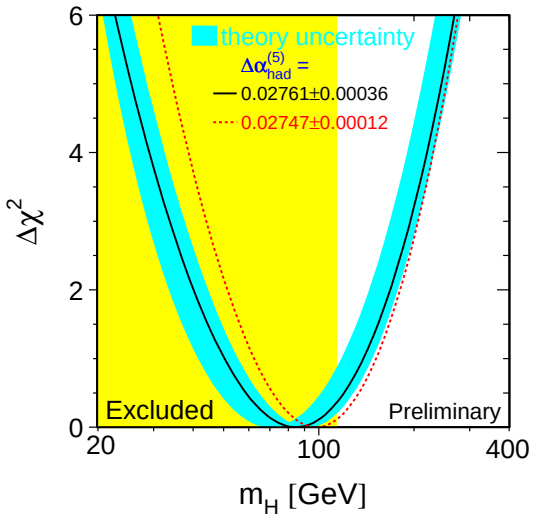


Figure 2.2: $\Delta\chi^2$ *distribution from the SM fit. The shaded area is excluded by direct observations.*

- spontaneous symmetry breaking: the Higgs mechanism has not been verified, the Higgs boson has not yet been discovered;

- large number of parameters: g_1, g_2, g_3, θ_{QCD} (the imaginary part of the strong coupling constant), m_h, m_Z, 6 quark masses, 3 lepton masses, 3 CKM mixing angles and 1 CKM phase;

- neutrino mass: neutrinos are exactly massless in the SM, nevertheless, from the observation of the neutrino flavour change, there is a compelling evidence that neutrinos have non-zero mass;

- unification of strong and electro-weak coupling constants: coupling constants change as a function of the energy scale of the process observed. It is believed that g_1, g_2, g_3 are the results of the splitting of a unique coupling constant of a unique force at very high energies; this fact cannot take place in the SM framework;

- hierarchy problem: to obtain a light Higgs mass ($O(10^2)$ GeV/c^2) it is necessary an unnatural fine tuning of the parameters;

- flavour mixing and the number of generations are arbitrary.

Solutions to these problems cannot be found in the SM, which is believed to be an effective theory valid only at low energies. New theories have been

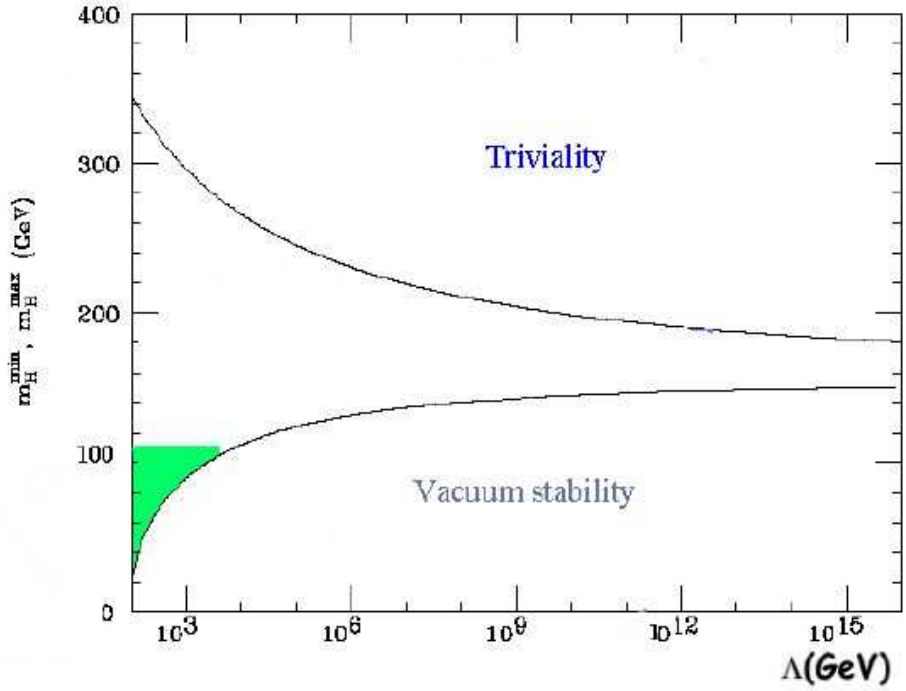


Figure 2.3: *Theoretical limits on the Higgs mass as a function of the energy scale* Λ. *The shaded area represents the values excluded by the direct search.*

introduced to forecast what could be the scenario at higher energies ($O(TeV)$ and higher) but they all still need to be verified. The most likely is supersymmetry.

2.2 Supersymmetry

The basic idea of a supersymmetric theory (SUSY) is the existence of a symmetry between fermions and bosons; for each fermion it exists a bosonic partner and vice-versa. Since such a symmetry has never been observed, it should be broken at a $O(TeV)$ energy, in order to leave the place to the SM low energy world. Moreover, for the same reason, the supersymmetric partners of the SM particles must be heavier than any particle which can have been produced at the modern accelerators.

2.2.1 Motivations for supersymmetry

The main motivation for such a theory is the unification of the fundamental interactions existing in nature. In fact, from the renormalization group equations, it comes that the coupling constant are *running*, that is they depends on the energy of the process. In details

$$\frac{1}{\alpha_i(Q^2)} = \frac{1}{\alpha_i(\mu^2)} - \frac{b_i}{4\pi} log\left(\frac{Q^2}{\mu^2}\right)$$

where $\alpha_i = \frac{g_i^2}{4\pi}$ and Q^2 and μ^2 are energy scales. b_i is a vector very different in SM and in SUSY theories: in the SM it is

$$b_i = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -22/3 \\ -11 \end{pmatrix} + 3 \begin{pmatrix} 4/3 \\ 4/3 \\ 4/3 \end{pmatrix} + \begin{pmatrix} 1/10 \\ 1/6 \\ 0 \end{pmatrix}$$

while in the MSSM (*Minimal Supersymmetric extension of the Standard Model*) it becomes

$$b_i = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -6 \\ -9 \end{pmatrix} + 3 \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} + 2 \begin{pmatrix} 3/10 \\ 1/2 \\ 0 \end{pmatrix}.$$

The resulting *running* is sketched in fig. 2.4 (it is assumed that the supersymmetric particles are heavier than $O(TeV/c^2)$). It is evident that no unification is possible in the SM, while it is in its minimal supersymmetric extension. The energy scale at which this unification takes place is $M_{GUT} \sim O(10^{16}\ GeV)$, where GUT stands for *Grand Unification Theory*.

Another motivation for the introduction of SUSY is the solution of the *hierarchy* problem. In particle physics there are a few characteristic energy scales:

- M_h at which $SU(2)$ symmetry is broken;
- M_{SUSY} at which supersymmetry is broken;
- M_{GUT} at which electro-weak and strong interactions are split (unified);
- M_{Pl} at which the gravitational coupling constant is split from (unified to) α_{GUT} (the electro-weak and strong unified coupling constant).

The questions that remain open are why it exists such a *hierarchy* and how theory can preserve it. SUSY can answer the second one. In the SM, radiative corrections, for instance to the calculation of the Higgs mass, would make it increase until spoiling the existing *hierarchy*. As mentioned in the previous section, only a very accurate fine tuning of the coupling constants could avoid this mass growth. In SUSY instead, cancellations take place in a natural way,

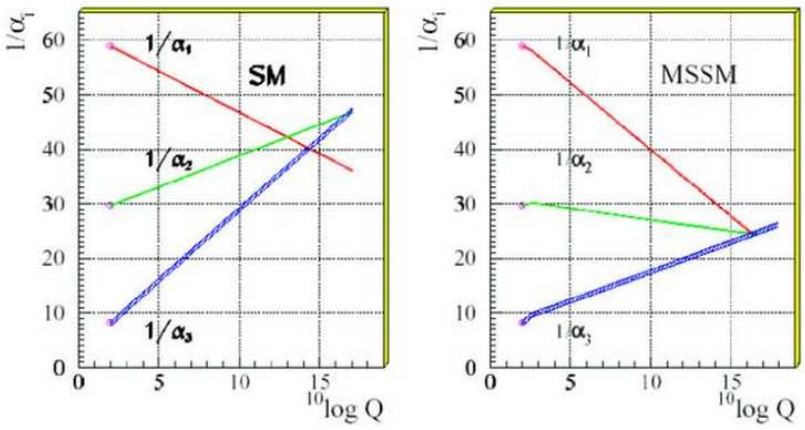


Figure 2.4: *Running of the electro-weak and strong coupling constants in the SM and in the MSSM.*

because of the mass degeneration of bosonic and fermionic partners at energies higher than M_{SUSY}:

$$\sum_{bosons} m^2 = \sum_{fermions} m^2.$$

At lower energies, it happens that

$$\sum_{bosons} m^2 - \sum_{fermions} m^2 = M_{SUSY}^2.$$

It follows that the corrections to M_h^2 are

$$\delta M_h^2 \sim g^2 M_{SUSY}^2 \sim M_h^2$$

if $M_{SUSY} \sim O(TeV/c^2)$, which is enough to avoid the spoiling of the *hierarchy*.

2.2.2 The MSSM

The MSSM is the *Minimal Supersymmetric extension of the Standard Model*. The SM is far from being a supersymmetric theory: bosons and fermions have completely different quantum numbers.

To supersymmetrize the SM it is necessary to introduce

- one new fermion for each SM boson,
- one new boson for each SM fermion,
- one Higgs boson doublet and two fermionic Higgs superpartners called higgsinos.

The MSSM set of particles is shown in table 2.5. A supersymmetric partner has been added for each known particle and a Higgs doublet has been implemented aside the SM one.

fermions		*bosons*		$SU(3) \bigotimes SU(2) \bigotimes U(1)$
gauge				
gluon	g^a	gluino	$\tilde{g}^a$	$(8,0,0)$
Weak	$W^k \quad (W^{\pm}, Z)$	wino, zino	$\tilde{w}^k \quad (\tilde{w}^{\pm}, \tilde{z})$	$(1,3,0)$
hypercharge	$B \quad (\gamma)$	bino	$\tilde{b} \quad (\gamma)$	$(1,1,0)$
matter				
sleptons	$\tilde{L}_i = (\tilde{\nu}, \tilde{e})_L$	leptons	$L_i = (\nu, e)_L$	$(1,2,-1)$
	$\tilde{E}_i = \tilde{e}_R$		$E_i = e_R$	$(1,1,2)$
squarks	$\tilde{Q}_i = (\tilde{u}, \tilde{d})_L$	quarks	$Q_i = (u,d)_L$	$(3,2,1/3)$
	$\tilde{U}_i = \tilde{u}_R$		$U_i = u_R$	$(3^*,1,-4/3)$
	$\tilde{D}_i = \tilde{d}_R$		$D_i = d_R$	$(3^*,1,2/3)$
Higgs				
Higgses	H_1	higgsinos	$\tilde{H}_1$	$(1,2,-1)$
	H_2		$\tilde{H}_2$	$(1,2,1)$

Table 2.5: *MSSM particles.*

In the MSSM it is always assumed that the so-called R-parity is conserved. The R-parity quantum number is defined by

$$R = (-1)^{3(B-L)+2S}$$

with B barion number, L lepton number and S spin of the particle. It follows that ordinary particles have $R = 1$, while supersymmetric partners have $R = -1$. As a consequence of R-parity conservation

- the interactions of superpartners are essentially the same as in the SM,
- the superpartners are created in pairs,
- the lightest superparticle (LSP) is stable.

Since such a particle has never been seen, it is assumed to be electrically neutral, heavy and weakly interacting with ordinary matter (the supersymmetric equivalent of neutrino).

2.2.3 Higgs in the MSSM

The super-potential of any supersymmetric theory has a zero vacuum expectation value. Applying this statement in the MSSM Higgs sector, it follows that no spontaneous SU(2) symmetry breaking can take place in this model. However, this is true only at energies at which supersymmetry is not broken, that is at the GUT scale. It happens that the parameters which occur in the Higgs potential depend on the energy scale, accordingly to the solutions of renormalization group equations. The *running* of these parameters can modify the vacuum expectation value of the Higgs potential; in fact it becomes non-zero at the electro-weak energy scale. This mechanism explains in a natural way the SU(2) symmetry breaking by mean of the radiative corrections; for this reason it is called radiative spontaneous symmetry breaking.

As mentioned in the previous section, the MSSM requires two Higgs doublets:

$$H_1 = \begin{pmatrix} H_1^0 \\ H_1^- \end{pmatrix} \qquad H_2 = \begin{pmatrix} H_2^+ \\ H_2^0 \end{pmatrix}.$$

In order to compute the mass spectrum it is useful to write them as

$$H_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} v_1 + h_1^0 + ia_1^0 \\ h_1^- \end{pmatrix} \qquad H_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} h_2^+ \\ v_2 + h_2^0 + ia_2^0 \end{pmatrix}$$

where v_1 and v_2 are the vacuum expectation values (*vev's*). The mass eigenstates can be derived from the previous fields by diagonalizing the mass matrices (see table 2.6). G^0 and $G^\pm$ are Goldstone bosons, while $h^0, H^0, A^0, H^\pm$ are the physical Higgs bosons.
The Higgs bosons' masses are

$$m_A^2 = m_1^2 + m_2^2$$

$$m_{H^\pm}^2 = m_A^2 + m_W^2$$

$$m_{H,h} = \frac{1}{2}\left(m_A^2 + m_Z^2 \pm \sqrt{(m_A^2 + m_Z^2)^2 - 4m_A^2 m_Z^2 cos^2 2\beta}\right)$$

where $m_{1,2}$ are the diagonal terms in the Higgs mass matrix and the β parameter is defined by

$$tan\beta = \frac{v_2}{v_1}.$$

$SU(2)_L$ eigenstates	mass eigenstates	
h_1^0, h_2^0	h^0 H^0	$CP - even,\ m_h < m_H$
a_1^0, a_2^0	A^0 G^0	$CP - odd$ $Goldstone : gives\ mass\ to\ Z$
$h_1^\pm, h_2^\pm$	$H^\pm$ $G^\pm$	$charged$ $Goldstone : give\ mass\ to\ W^\pm$

Table 2.6: *Higgs fields.*

In the MSSM the Higgs and the electro-weak bosons masses satisfy the following relations:

$$m_{H^\pm} \geq m_W$$

$$m_h \leq m_A \leq m_H$$

$$m_h \leq m_Z |cos2\beta| \leq m_Z$$

$$m_h^2 + m_H^2 = m_A^2 + m_Z^2.$$

The fact that the lighter Higgs boson mass is less than the Z one is in contradiction with the experimental observations, which have set a lower limit on the Higgs mass at about 114.4 GeV/c^2. However, applying the higher orders corrections, the theoretical higher limit raises up till about 125 GeV/c^2 (depending on the particular set of MSSM parameters used).

2.3 mSUGRA

Spontaneous supersymmetry breaking cannot take place by mean of the MSSM fields only. It is necessary to introduce a so-called hidden sector which breaks

SUSY and witch communicates this breaking to the MSSM by mean of some messenger.

In gravity-mediated SUSY braking models (SUGRA), gravity is the only messenger. The minimal supergravity model (mSUGRA) assumes that, at the GUT scale, all squarks, sleptons and Higgs bosons have a common mass m_0, that all gauginos and Higgsinos have a common mass $m_{\frac{1}{2}}$ and that all trilinear Higgs-sfermion-sfermion couplings have a common value $\AA_0$. The bilinear SUSY breaking term B and the SUSY conserving Higgs mass μ are the remaining free parameters; moreover, they can be traded in for the Higgs vev's (v_1 and v_2), with a two fold ambiguity in $sgn(\mu)$.

The mSUGRA parameters are therefore:

$$m_0 \quad m_{\frac{1}{2}} \quad A_0 \quad tan\beta \quad sgn(\mu)$$

2.4 The A boson search

The *CP-odd* neutral Higgs boson A^0 has been searched for at the LEP experiments without success. Nevertheless, a new lower mass limit has been calculated ($m_A > 91.9$ at 95% CL [4]) and new exclusion regions have been drawn. In fig. 2.6 the combined LEP results are plotted in the $tan\beta$-m_A plan (the so-called *max-mixing* scenario has been chosen).

Near-to-come LHC experiments are expected to discover, if they exist, all the supersymmetric particles. In the following section the way the A^0 boson could be produced and discovered at LHC is described. The framework of the ATLAS experiment has been used.

2.4.1 The A boson at the ATLAS experiment

At the LHC, the A boson is produced through the gg fusion directly (see fig. 2.7) or with the association of a $b\bar{b}$ pair (see fig. 2.8). The associated production is dominant at high $tan\beta$ values, while is less important for low ones. In fig. 2.9 [5] are sketched the production cross-section at four different $tan\beta$ values as a function of the A boson mass.

The main characteristics of the A boson branching ratios can be summarized as follow (see fig. 2.10, 2.11, 2.12):

- WWA, ZZA and hhA couplings are absent: there are no ZZ, WW or hh decay channels;

- at low $tan\beta$, $t\bar{t}$ is the dominant decay at masses higher than the kinematic threshold, while at lower masses Zh and $\tau\tau$ channels are the most relevant;

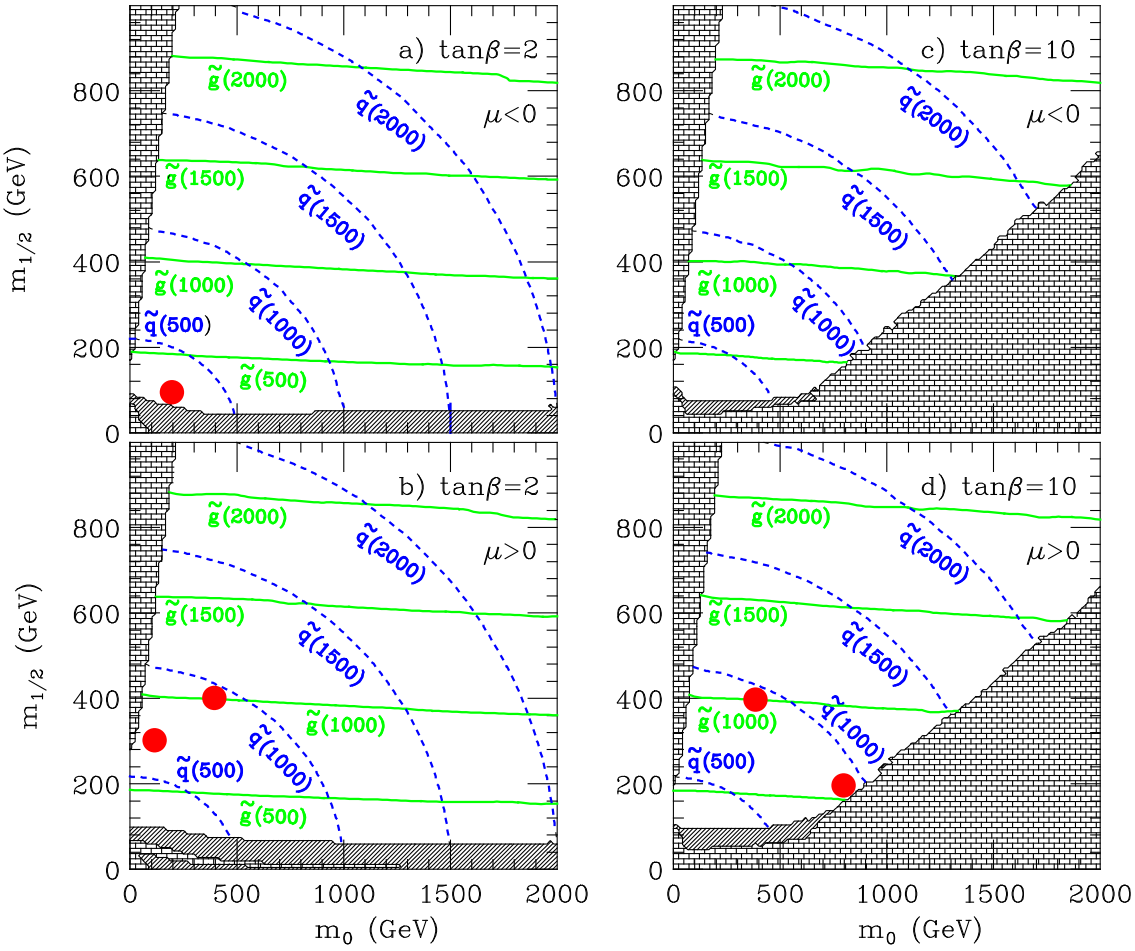


Figure 2.5: *Gluino and squark masses in the m_0 and $m_{\frac{1}{2}}$ at $A_0 = 0$ and with two different tanβ and μ values. The bricked regions at small m_0 are excluded by requiring that $\tilde{\chi}_1^0$ is the LSP. The bricked regions at large m_0 and tanβ are excluded by having no symmetry breaking. The cross-hatched regions are excluded by the experiment.*

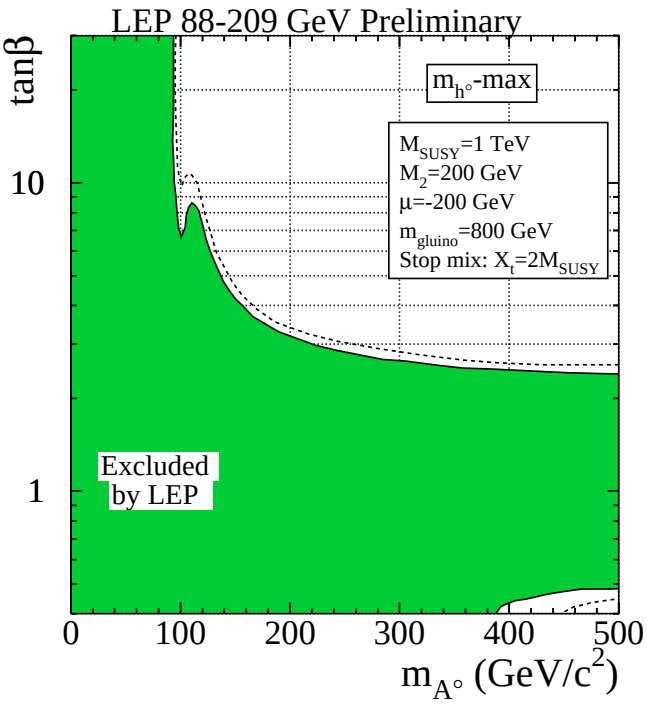


Figure 2.6: *$m_A - tan\beta$ plan: the dark region is excluded experimentally.*

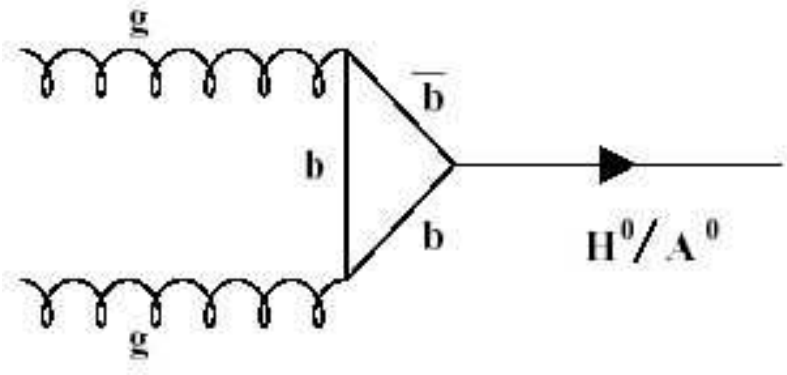


Figure 2.7: *Direct production.*

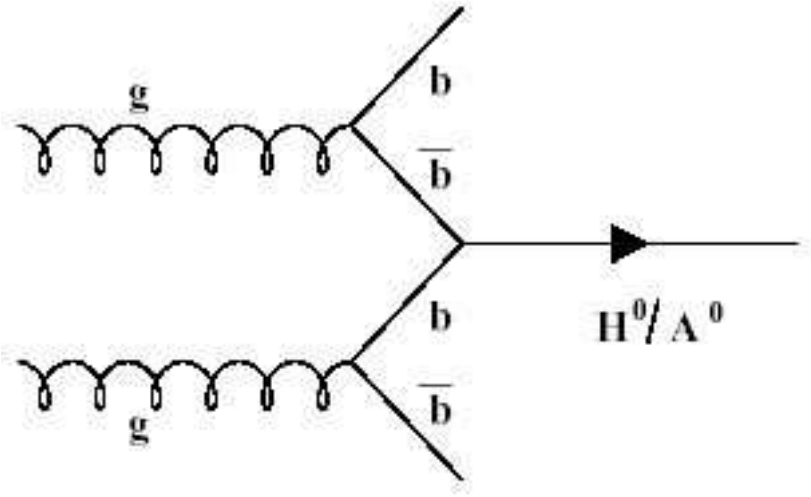


Figure 2.8: *Associated production.*

- at high $tan\beta$ the dominant decay mode is $b\bar{b}$, with a $\tau\tau$ contribution around the 10%; the $A \to \mu\mu$ branching ratio is suppressed by a factor $(m_\mu/m_\tau)^2$ w.r.t. the $A \to \tau\tau$ one.

The expected decay rates are summarized in tab. 2.7, 2.8, 2.9 and 2.10 where $\sigma \times BR$ of the most contributing channels is shown at different $tan\beta$ and m_A values.

m_A (GeV)	$tan\beta = 3$	$tan\beta = 10$	$tan\beta = 30$
150	1.05	1.97	18.8
300	0.12	0.15	1.47
500	0.001	0.02	0.15

Table 2.7: *Expected rates for $A \to \tau\tau$ (in pb).*

Further details about the $A \to \tau\tau$ channel will be given in chapter 4.

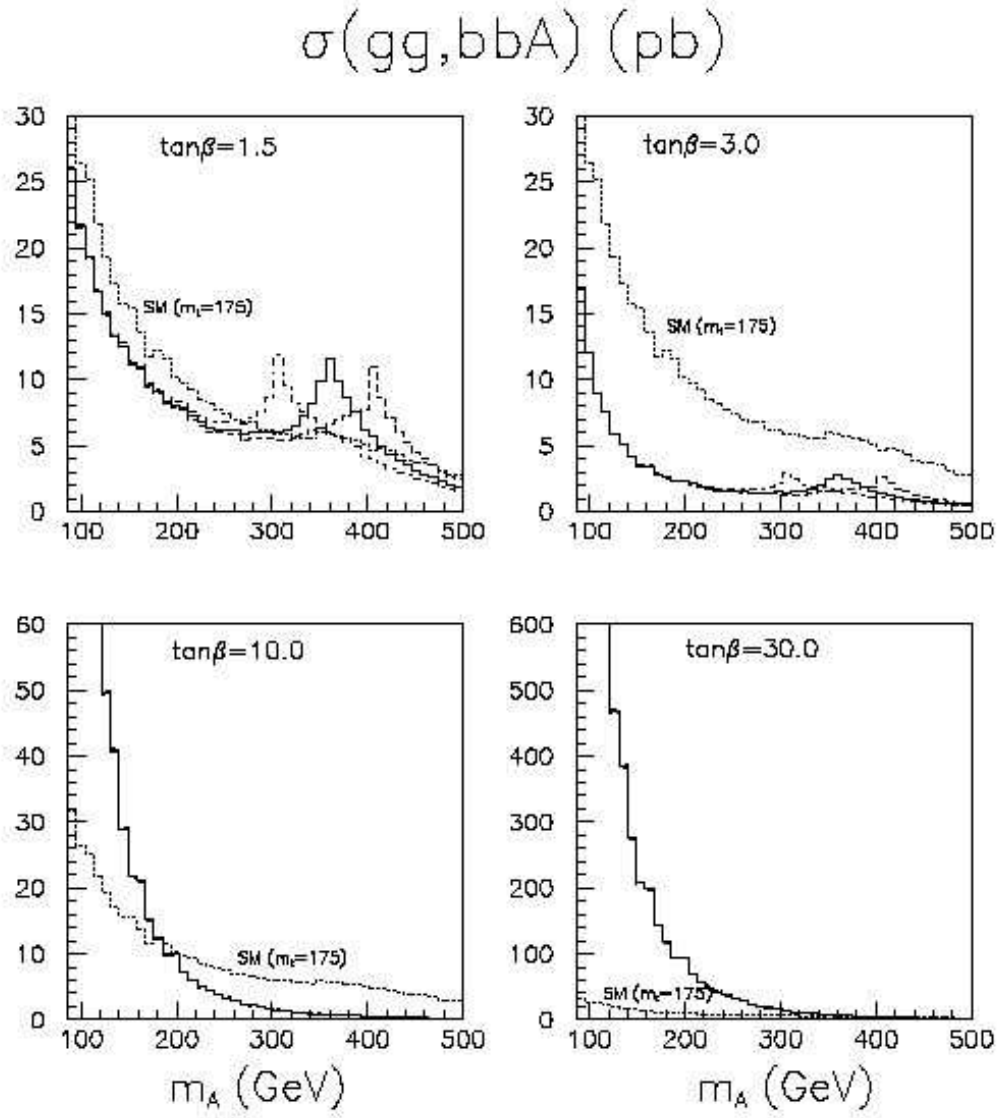


Figure 2.9: *Production cross-section for an A-boson ($gg \to A$, $b\bar{b}A$) as a function of mA for four values of tanβ. The solid line is for $m_t = 175$ GeV, the dashed one for $m_t = 200$ GeV and the dot-dashed one for $m_t = 150$ GeV. The figures also show the SM predictions for $m_t = 175$ GeV (dotted lines).*

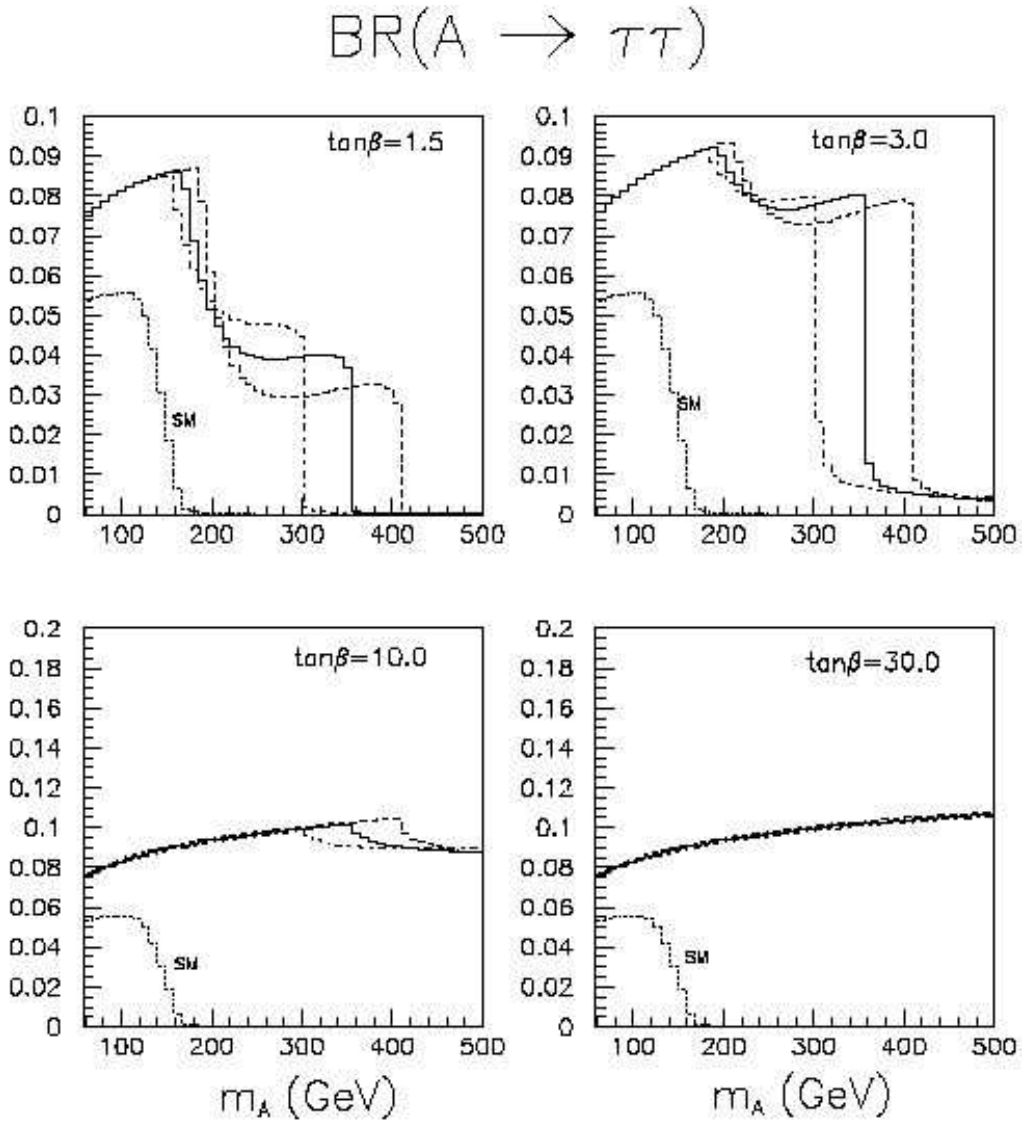


Figure 2.10: *$A \to \tau\tau$ branching ratio as a function of m_A for four values of tanβ. The solid line is for $m_t = 175$ GeV, the dashed one for $m_t = 200$ GeV and the dot-dashed one for $m_t = 150$ GeV. The figures also show the SM predictions (dotted lines).*

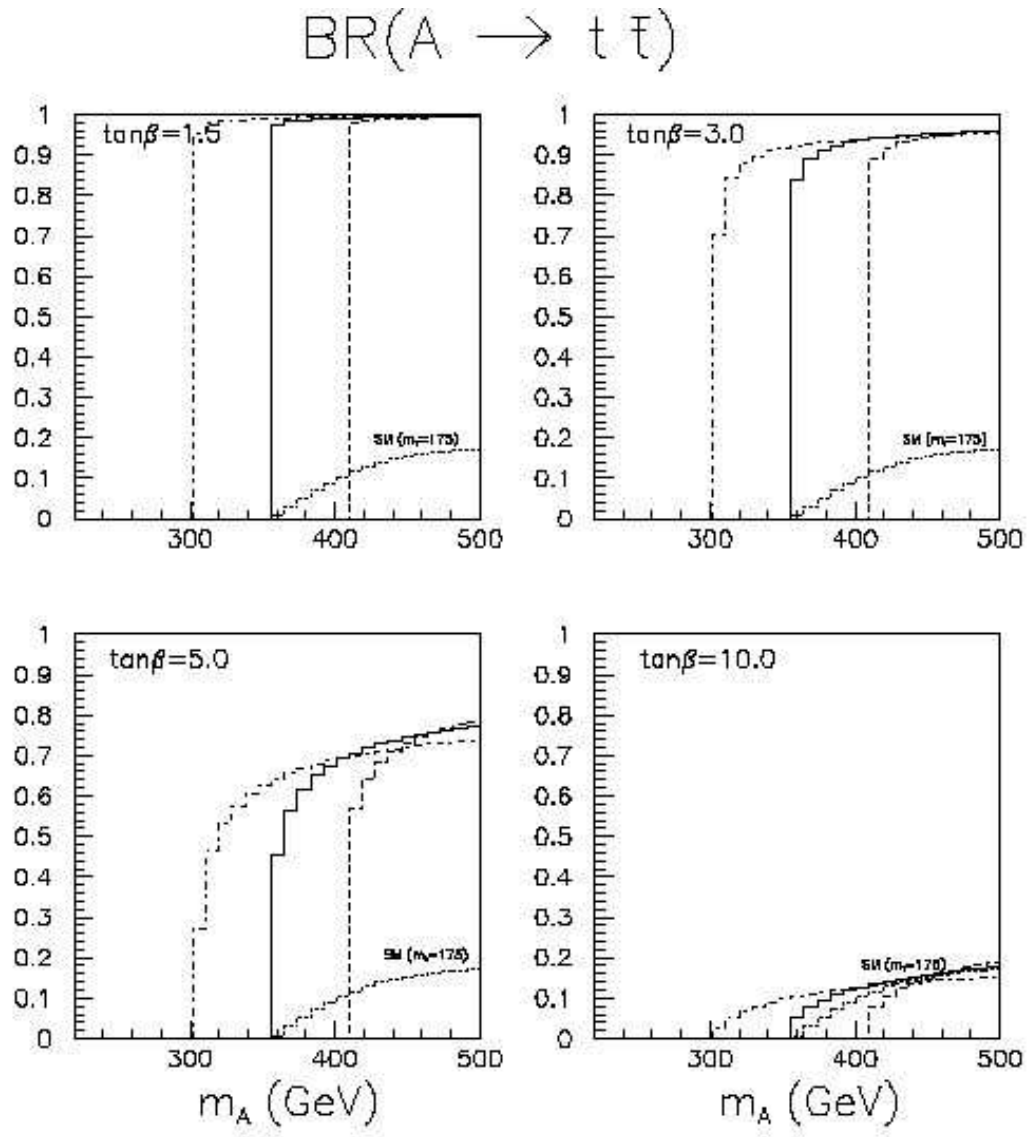


Figure 2.11: *$A \to t\bar{t}$ branching ratio as a function of m_A for four values of tanβ. The solid line is for m_t = 175 GeV, the dashed one for m_t = 200 GeV and the dot-dashed one for m_t = 150 GeV. The figures also show the SM predictions (dotted lines).*

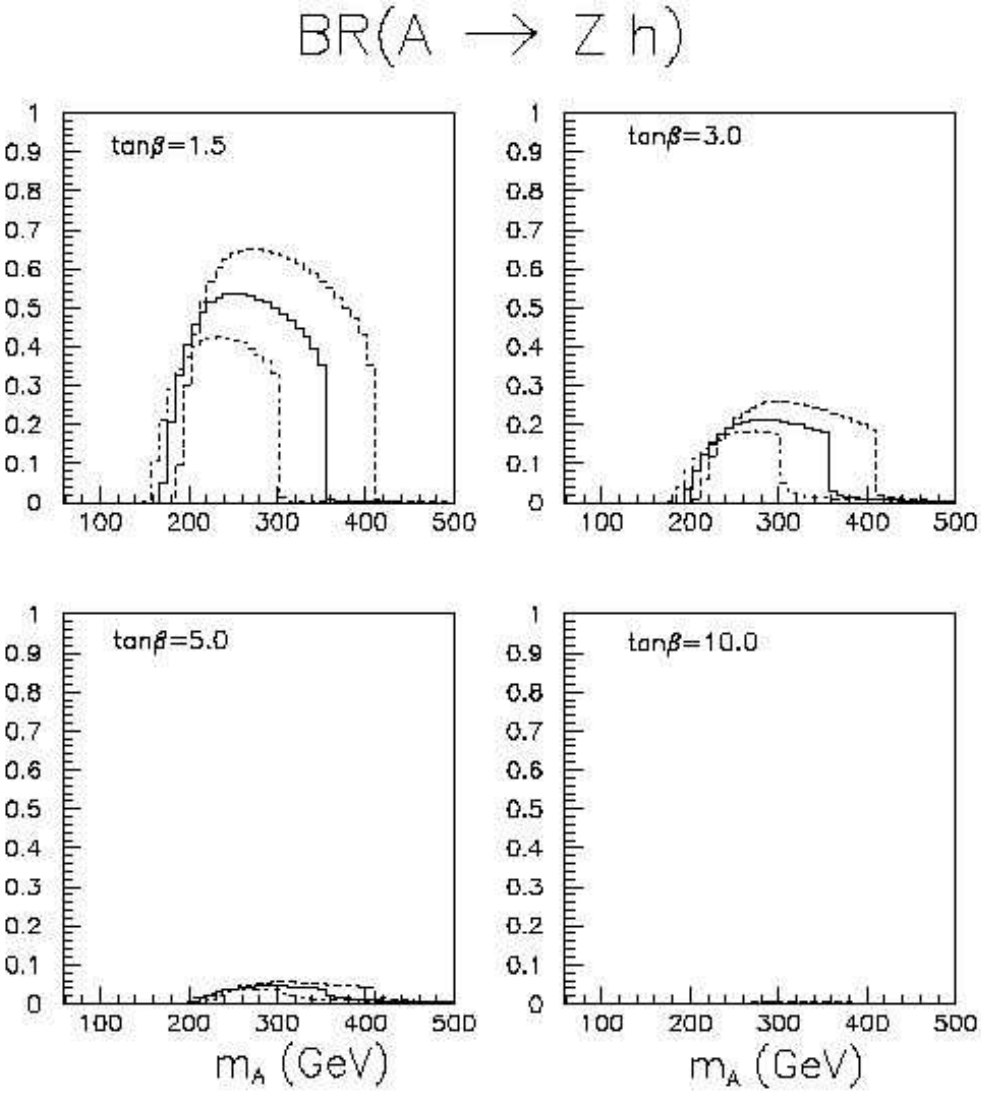


Figure 2.12: *$A \rightarrow Zh$ branching ratio as a function of mA for four values of $tan\beta$. The solid line is for $m_t = 175$ GeV, the dashed one for $m_t = 200$ GeV and the dot-dashed one for $m_t = 150$ GeV.*

$m_A\ (GeV)$	$tan\beta = 3$	$tan\beta = 10$	$tan\beta = 30$
150	0.32	6.81	65.0
300	0.12	0.53	5.08
500	0.003	0.06	0.54

Table 2.8: *Expected rates for* $A \to \mu\mu$ *(in* fb*).*

$m_A\ (GeV)$	$tan\beta = 3$	$tan\beta = 5$	$tan\beta = 10$
400	1.51	0.45	0.03
450	0.80	0.27	0.02
500	0.25	0.15	0.02

Table 2.9: *Expected rates for* $A \to t\bar{t}$ *(in pb).*

$m_A\ (GeV)$	$tan\beta = 3$	$tan\beta = 5$	$tan\beta = 10$
250	21	2.65	0.55
300	17	1.94	0.29
350	31	2.54	0.14

Table 2.10: *Expected rates for* $A \to Zh$ *(in* fb*).*

Bibliography

[1] J.F.Gunion et al.
The Higgs hunter's guide
Addison-Wesley Publishing Company - 1990

[2] H.Haber
Lectures at the Summer School of Particles Physics
The "Abdus Salam" International Centre for Theoretical Physics

[3] The LEP Collaborations ALEPH, DELPHI, L3, OPAL, the LEP Electroweak Working Group, and the SLD Heavy Flavour Group
A Combination of Preliminary Electroweak Measurements and Constraints on the Standard Model
Prepared from Contributions of the LEP and SLD Experiments to the 2002 Winter Conferences.

[4] The LEP Collaborations ALEPH, DELPHI, L3, OPAL, the LEP Higgs Working Group
Search for the Standard Model Higgs Boson at LEP
LHWG Note/2002-01, July 2002

[5] Elzbieta Richter-Was, Daniel Froidevaux, Fabiola Gianotti, Luc Poggioli, Donatella Cavalli, Silvia Resconi
Minimal Supersymmetric Standard Model Higgs rates and backgrounds in ATLAS
ATLAS Internal Note PHYS-No-074 22/04/1996

[6] D.Cavalli et al.
The Higgs Working Group Summary Report
hep-ph/0203056 - 5 Mar 2002

[7] The ATLAS Collaboration
ATLAS Detector and Physics Performance Technical Design Report
CERN - LHCC 99/14-15

[8] S.Ferrara
Supersymmetry
North-Holland 1987

[9] P.West
Introduction to Supersymmetry nas Supergravity
World Scientific Publishing 1990

[10] R.N.Mohapart
Unification and Supersymmetry
Springer-Verlag 1992

[11] H.J.W.Muller-Kirsten
Supersymmetry
World Scientific 1987

Chapter 3

Vertexing

3.1 Generalities

In high-energy physics events, the particles measured in the tracking detectors mainly emerge from the primary interaction, from a particle decay or from a nuclear interaction in the detector. In order to understand the event topology and kinematics, it is important spotting the points (vertices) where the particles have had their origin.
In collider physics, vertices are usually classified as follows:

- **primary vertex**: the point where the two incoming beam particles (e^+e^-, $p\bar{p}$, pp, etc.) have collided;
- **secondary vertex**: the point where a particle is decayed or where it has undergone a nuclear interaction.

Since unstable particles mean-path is usually very short (O(mm)), very good spatial resolution is necessary to allow the separation of primary and secondary (decay) vertices. In order to improve the resolution, high-granularity detectors, to be placed near the interaction region, have been designed and built. This kind of detectors, such as the ATLAS Pixel apparatus (see chapter 1), can provide a very dense matrix of points, which allows a very accurate track reconstruction in this region and improves the vertexing resolution. This feature is fundamental in long-lived particles lifetime measurement which can simply be inferred by measuring the distance between the primary and the decay vertex. Moreover, good resolution is helpful in flavour tagging, which is the identification of a particular flavour in the event (usually b quarks or τ leptons).
A detailed description of a procedure to tag τ's is described in the next chapter.

Even if many strategies can be followed to find vertices, all the algorithms make basically a fit of tracks, recalculating their parameters, in order to force them to converge to a unique point in the space (the vertex). The basic idea is

to perform a fit of the tracks parameters, in order to find a common point to all tracks.

3.2 Pattern recognition and track fitting

The tracks are reconstructed by algorithms whose input are the digitized data coming from the detectors. To perform vertexing, it is necessary a good track reconstruction in the detectors near the interaction region and the ATLAS Inner Detector (see chapter 1) has been designed to achieve this goal.
Two algorithms (*iPatRec*[1] and *xKalman*[2]) are devoted to this task and follow two different reconstruction strategies. In this work only the second one has been used.

xKalman is aimed to reconstruct charged particles with $p_T > 0.5$ GeV/c. It starts the pattern recognition by finding track segments in the TRT and uses two different techniques to reconstruct tracks: a histogramming method and the Kalman filter-smoothing formalism [3].
The *xKalman* strategy can be summarized in three steps:

1 - global pattern recognition in the TRT using 2-D projections of the tracks in the R, ϕ (barrel) and z, ϕ (endcaps) plane. The resulting helices are used to define a track road through the precision tracker (SCT and Pixel planes).

2 - Using the Kalman filter-smoother formalism all the possible helix trajectories within the track road are looked for. Each track candidate is fitted and labeled with a quality flag; the candidates of insufficient quality are rejected.

3 - All the accepted helices are extrapolated back to the TRT; all the straw hits near the trajectory are included for the final track finding and track fitting steps.

A track is eventually accepted and stored if it satisfies a set of requirements, concerning the absolute number of precision (Pixel and SCT) hits, the number of precision hits not shared with other tracks, the p_T, the number of TRT hits and the fraction of straws hit.

The track is defined by the set of following parameters:

- the fit χ^2;
- the transverse impact parameter at the closest approach to the nominal vertex;
- the z-coordinate at the closest approach in $r - \phi$;
- the ϕ angle at the closest approach to the nominal vertex;

- the $cotg(\theta)$ at the closest approach to the nominal vertex;
- $1/p_T$ at the nominal vertex (negative for tracks with negative charge);
- the covariance matrix at the nominal vertex;
- the radius at the intercept with the Inner Detector outer boundary;
- the ϕ angle at the intercept with the Inner Detector outer boundary;
- the z-coordinate at the intercept with the Inner Detector outer boundary;
- the difference in ϕ between the ϕ direction at the intercept with the Inner Detector outer boundary and the ϕ at the point where the track leaves the Inner Detector;
- the $cotg(\theta)$ at the intercept with the Inner Detector outer boundary;
- $1/p_T$ at the Inner Detector outer boundary (negative for tracks with negative charge);
- the covariance matrix for parameters at the outer surface of the Inner Detector.

3.3 Vertexing

The reconstructed tracks are the vertex algorithm input. A selection on the tracks to be fitted in a common vertex is usually performed: as an example, the tracks candidate to belong to the primary/secondary vertex are preselected by the impact parameter value.
The fitting routines can use the track position and error matrix only (geometrical vertex fits) or also the track momentum (kinematical fits). Moreover, the fits can be constrained, i.e. the fitted tracks parameters will be constrained to satisfy some particular equation. For instance, particles invariant mass may have to be conserved or the resulting vertex momentum may have to point to another vertex.

The algorithm which is used in the following has been taken from the CDF experiment [4] and performs geometrical fits (if no constraints are imposed), based on a χ^2 minimization method.

The minimization starts from the point which is the intersection of the first two tracks to fit. Its position is found projecting the two helices onto the r, ϕ plane: if the resulting circles intersect, the point with the smaller z difference between the two helices (extrapolated to the intersection point) is chosen (see fig. 3.1); if they do not intersect, the vertex approximation is taken as the point of closest approach of the two circles.

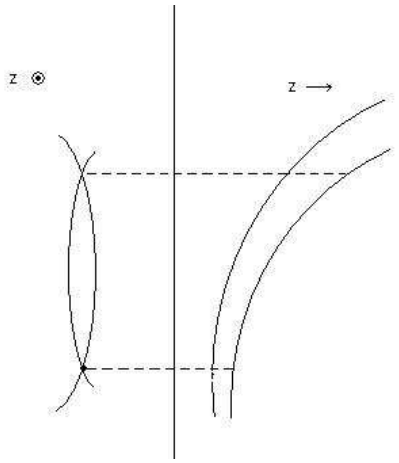


Figure 3.1: *First vertexing approximation from two tracks.*

The routine finds an initial approximation to the track parameters and adjust them in order to minimize the χ^2. The minimization proceeds by expanding the equations in a Taylor series around the last approximation, keeping only the linear terms and solving the resulting linear equations. The expansion and solution is iterated several times. The iteration is repeated until the linear terms become negligible small [5].

In this algorithm the fit procedure can be iterated. The mechanism (see fig. 3.2) allows at each step the deletion of the track with the higher contribution to the overall vertex χ^2, in order to eliminate from the fit the tracks not belonging to the vertex. The final χ^2 cut has to be optimized to allow the deletion of as many wrong tracks as possible but also to avoid the rejection of tracks pointing correctly to the vertex.

3.4 The software

In order to allow advanced code writing, the ATLAS experiment has chosen to develop the code using object oriented strategies. In particular, the C++ has been chosen as the main collaboration language, raising the problem of merging the new code with that written before. This is the case of the vertex fitting routine core, written in FORTRAN and therefore wrapped by a C++ interface.

The interface aim is to provide tools to allow the user to dialog with the FORTRAN fitting program by mean of C++ methods. These methods handle the fit I/O, put and retrieve the tracks and vertices parameters, set the constraints

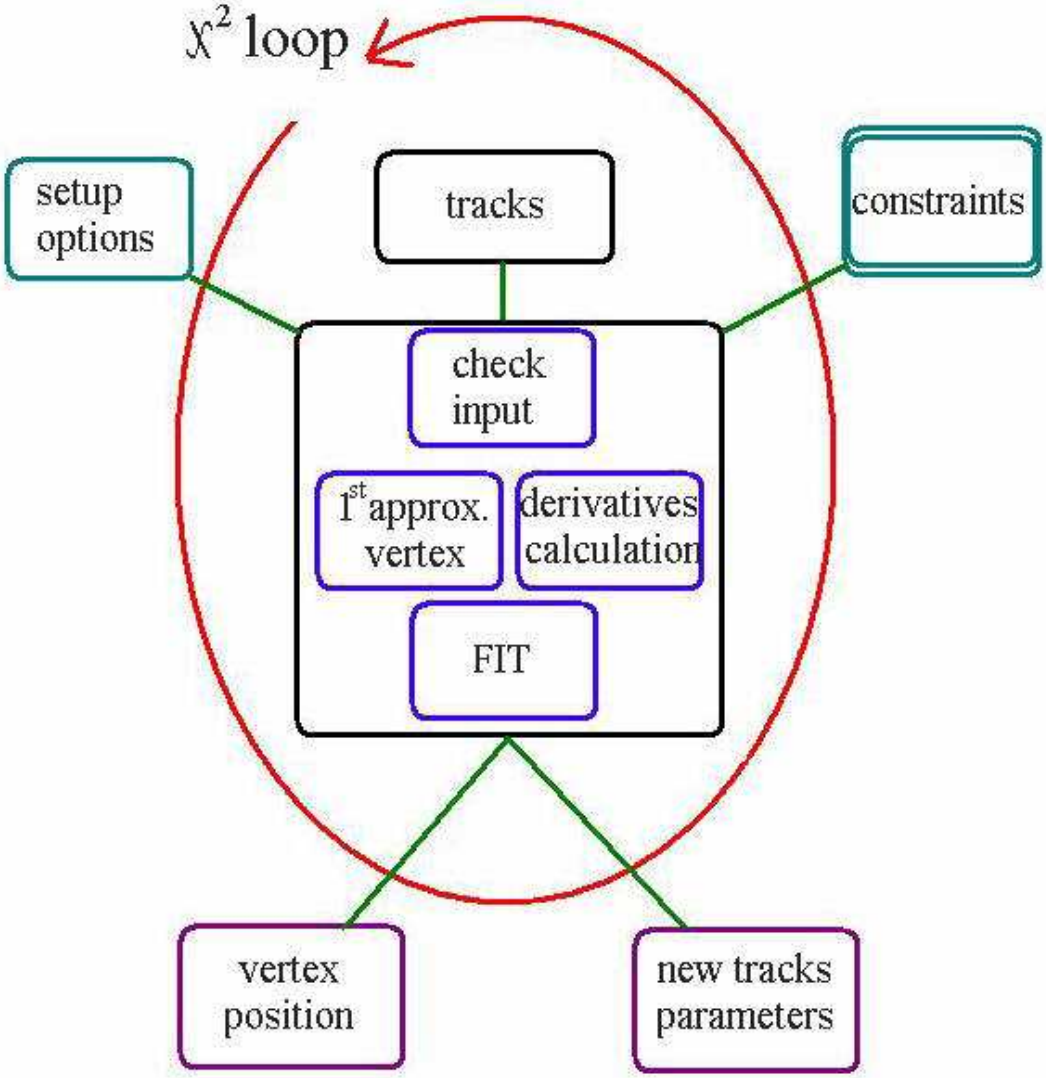


Figure 3.2: *A summary of the vertexing procedure.*

and manage the deletion of the tracks with the highest χ^2 contribution.
The interface can operate in three different modes:

- **mode 1**: only the primary vertex is fitted. All the reconstructed tracks which satisfy a few basic quality cuts are passed to the vertexing; the χ^2 control provides a further selection to eliminate tracks emerging from secondary vertices.

- **mode 2**: only the secondary vertices are fitted; the primary vertex position and error matrix are hardwired.

- **mode 3**: both primary and secondary vertices are reconstructed.

In the case of secondary vertices, the track selection is strongly dependent on the physical process at study. The list of the tracks to be fitted must be created by the specific user algorithm and sent to the interface as input.
In order to show the interface tools and to test their capabilities, a few algorithms have been written and tested on simulated data.

A detailed description of the vertexing packages and of their software technicalities is provided in appendix A.

3.5 The tests

The *VertexPrimary* algorithm, used to test the primary vertex reconstruction, works with the interface set to operate in mode 1. All the reconstructed tracks that satisfy a few quality cuts are used to fit the primary vertex. By default they are:

- impact parameter smaller than 300 μm

- at least 9 precision hits

- at least 1 b-layer hit

- at least 2 pixel hits

- transverse momentum (p_T) greater than 500 MeV

A sample of about 800 events with a 100 GeV/c^2 SM Higgs decaying into $b\bar{b}$ has been used to test the algorithm. In figs. 3.3 and 3.4, the resolutions with and without the χ^2 loop are shown respectively. The resolution is improved using the loop, since, at the end of the iteration, most of the tracks emerging from the B secondary vertices have been dropped.
The resulting RMS's are summarized in table 3.1.

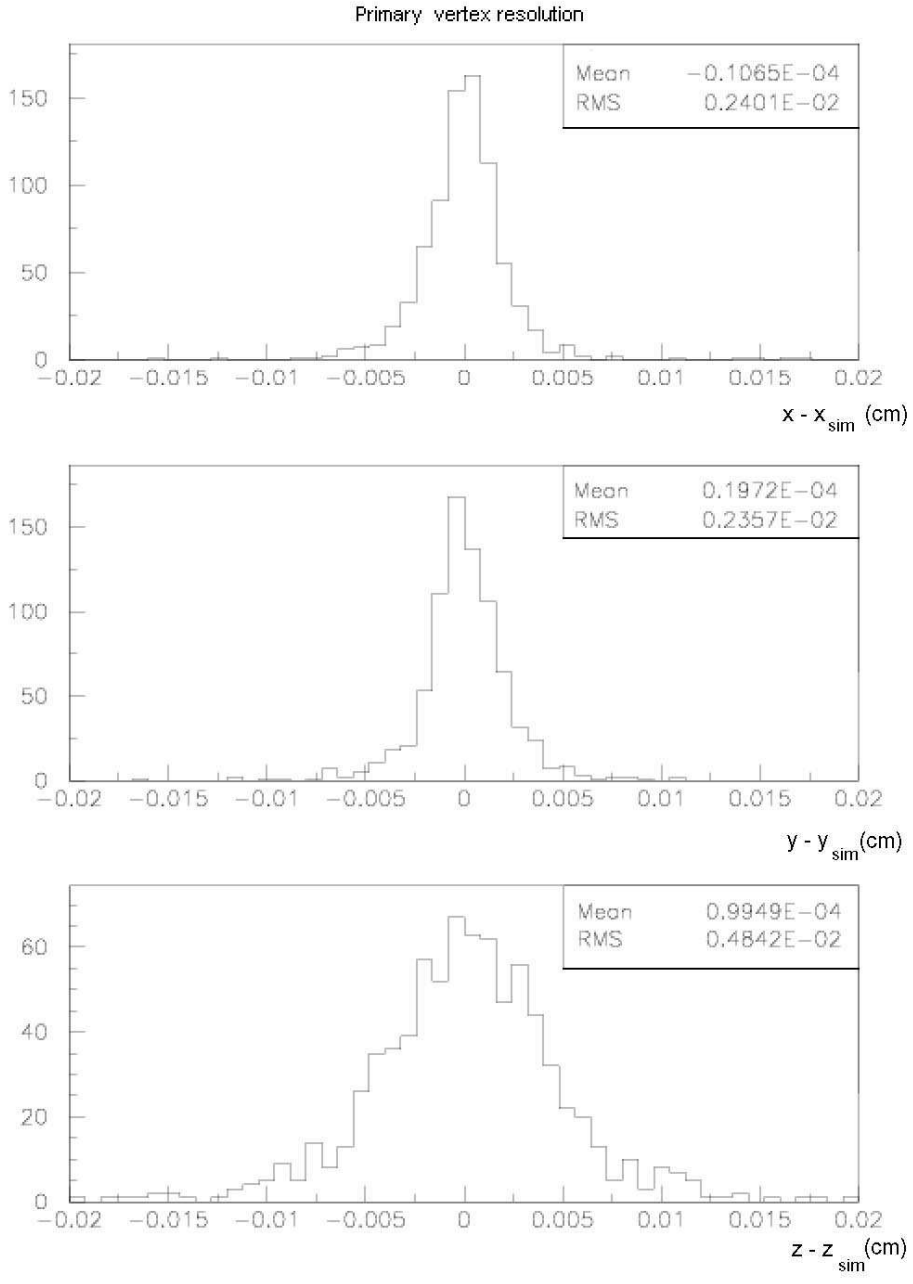


Figure 3.3: *Primary vertex resolution plots.* χ^2 *loop activated.*

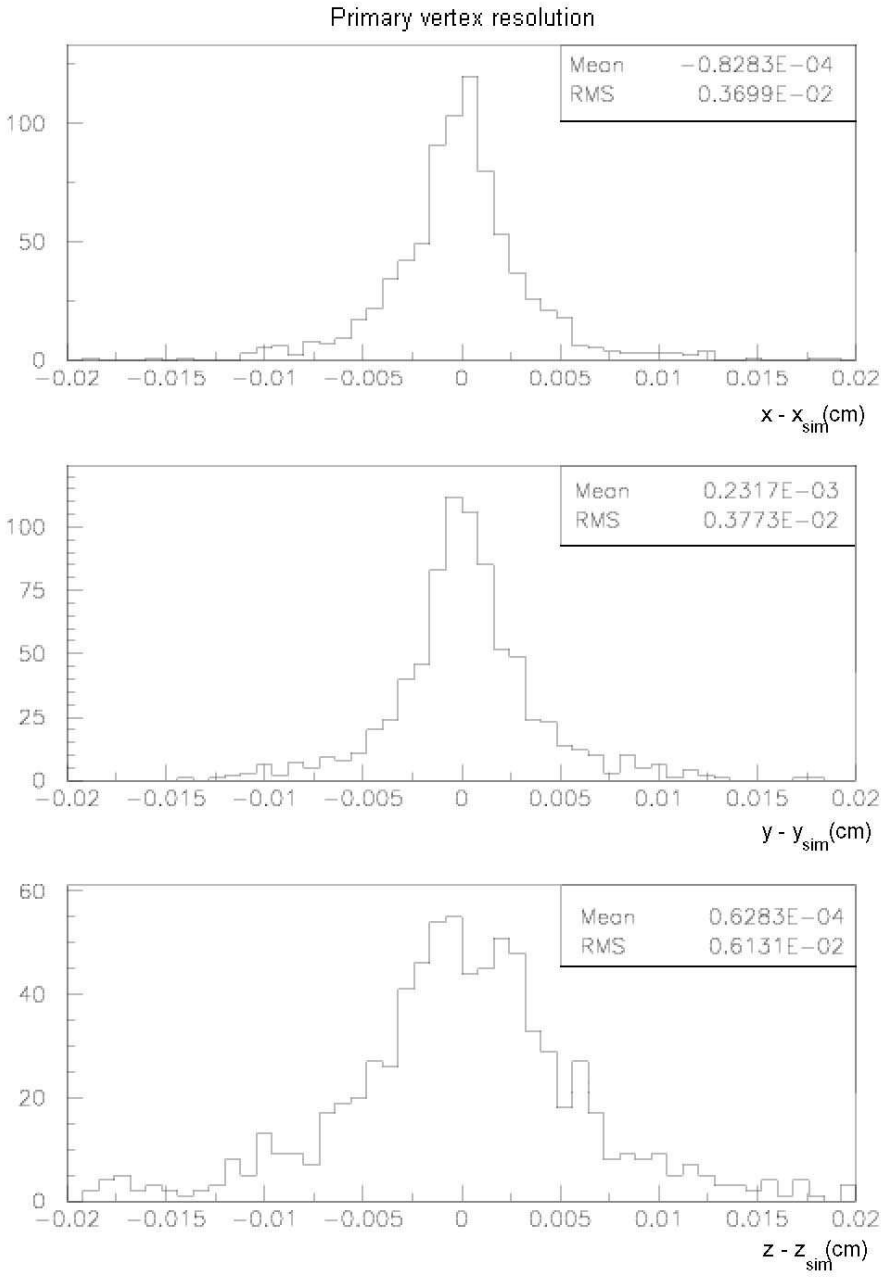


Figure 3.4: *Primary vertex resolution plots. χ^2 loop not activated.*

RMS's (μm)			
	x	y	z
Loop enabled	24	24	48
Loop disabled	37	38	61

Table 3.1: *Primary vertex resolutions ($H \rightarrow b\bar{b}$ events).*

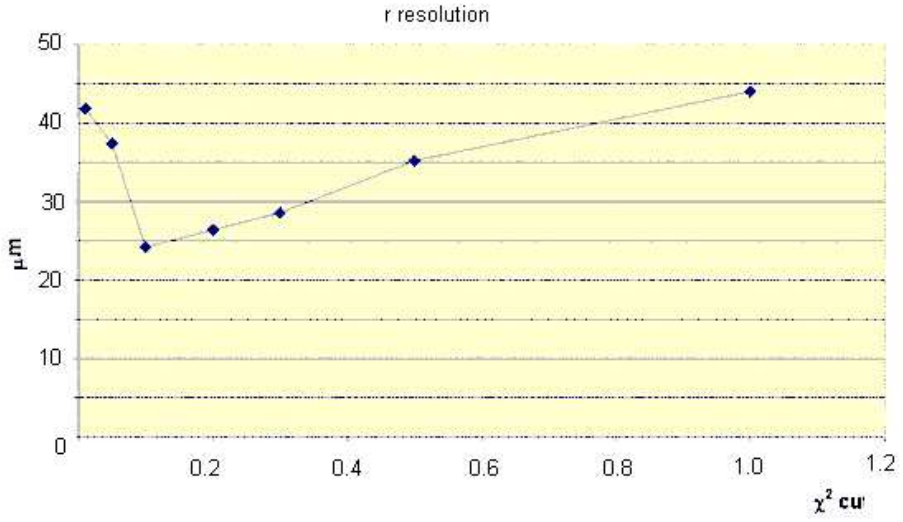


Figure 3.5: *Primary vertex resolution in the $r - \phi$ plane as a function of the χ^2 cut ($H \rightarrow b\bar{b}$ events).*

The maximum χ^2 contribution allowed for a track has been chosen performing a scan and looking for a minimum of the resolution. As shown in fig. 3.5, the minimum is reached at $\chi^2 = 0.1$. A higher χ^2 cut yields a worse resolution since some of the tracks from the secondary vertices are not rejected; on the other hand, with a higher χ^2 cut, the vertex fit loses precision. The number of tracks fitted in the primary vertex as a function of the χ^2 cut is shown in fig. 3.6.

The *VertexExampleSecSeed* algorithm has been used to test the interface technicalities when reconstructing secondary vertices and has been roughly tuned to search for B decay vertices. The same 100 GeV/c^2 Higgs events has been used as data sample.
In this example, a b quark is looked for in the Pythia simulated data (GENZ) and its direction is used as a seed. All the tracks satisfying a few quality cuts and contained in a defined amplitude cone around the seed are used in the secondary

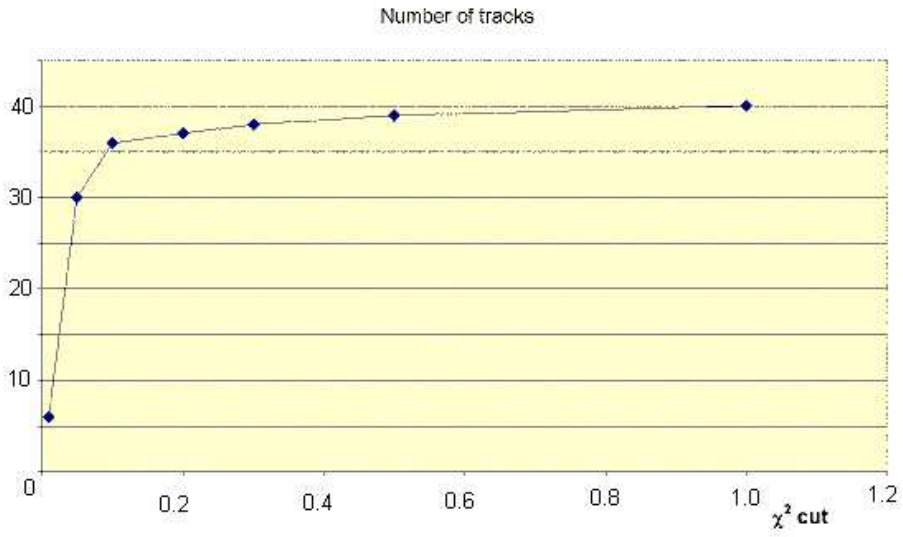


Figure 3.6: *Number of tracks used in the primary vertex fit ($H \rightarrow b\bar{b}$ events).*

vertex fit. The following selection cuts on the input tracks have been applied:

- impact parameter less than 1 mm and greater than 10 μm
- at least 9 precision hits
- at least 1 b-layer hit
- at least 2 pixel hits
- transverse momentum greater than 500 MeV
- cone amplitude around the seed $\Delta R = \sqrt{\eta^2 + \phi^2} = 1.4$

No mass constraints have been applied.
The cuts on the tracks quality are the same used in the *VertexPrimary* algorithm, while the cut on the impact parameter allows larger Impact Parameter values and at the same time excludes most of the tracks coming from the interaction region. The interface has operated in mode 3 and the χ^2 loop has been disabled. The resolution achieved is shown in table 3.2 and in fig. 3.8.

Since both primary and secondary vertex position is available, a rough estimation of the B mesons mean life can be attempted. In fig. 3.7 the exponential fit result of the primary-secondary vertex distance distribution (projected onto the $r - \phi$ plane) is shown. The fitted value of $\gamma c\tau$ is (3.3 ± 0.3) mm (the error is underestimated since the vertexing resolution is not included), to be compared

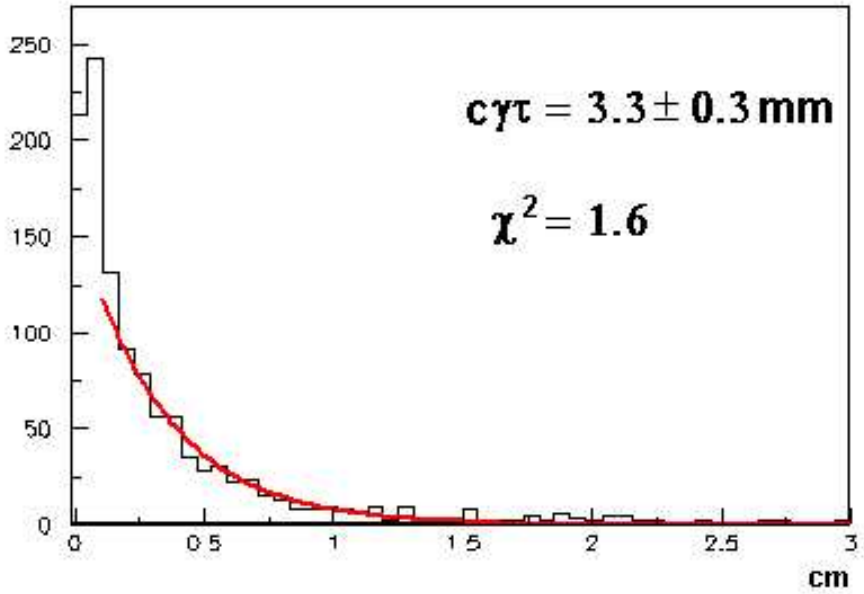


Figure 3.7: *$\gamma c\tau$ for B's (the vertexing resolution is not included in the error estimation).*

with the theoretical expectation for a B mixture: $\gamma c\tau \simeq 4.3\ mm$ (an average transverse B momentum of 50 GeV/c has been considered).

RMS (cm)		
x	y	z
0.32	0.41	0.56

Table 3.2: *B vertex resolution.*

The *VertexExampleSecUserTracks* algorithm has been written in order to test the possibility of selecting a user-made list of tracks. A sample of about 5000 $A \to \tau\tau$ events have been analyzed.
The χ^2 cut for the primary vertex fit has been optimized. The results (see figs. 3.9, 3.10) are quite similar to those obtained with the $H \to b\bar{b}$ data sample.

The track list for the τ vertex fit has been created storing the tracks associated to a three prongs hadronic τ decay (see chapter 4 for details). The τ vertex resolution is shown in table 3.3.

As well as for B's, an approximated mean life estimation has been performed. The result is shown in fig. 3.11. The estimation is in fair agreement

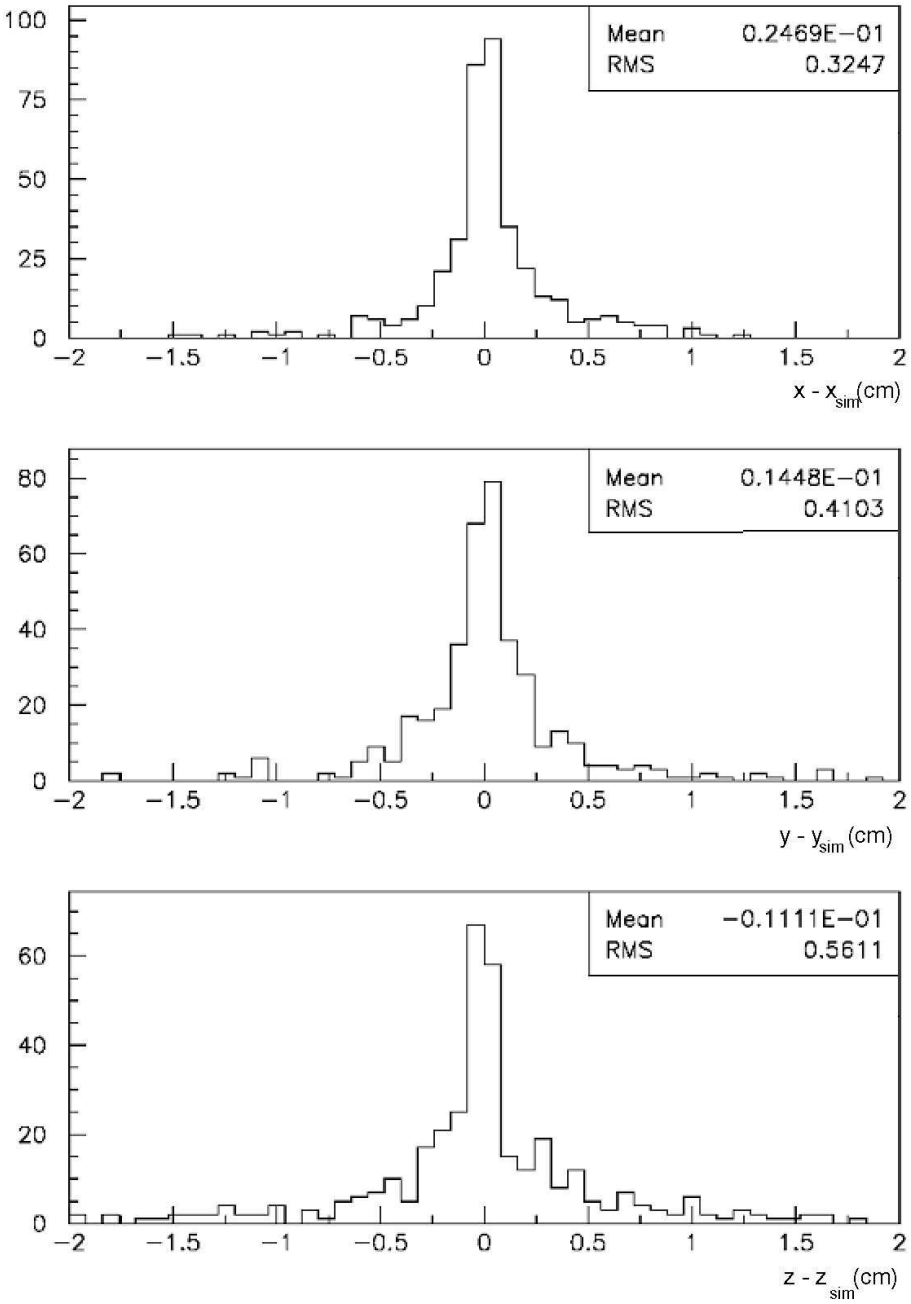


Figure 3.8: *B vertices resolution plots. χ^2 loop disabled.*

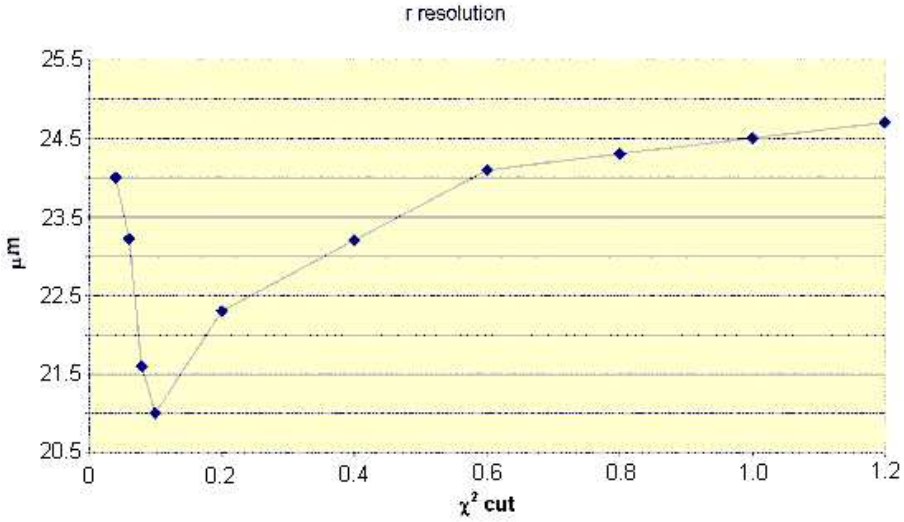


Figure 3.9: *Primary vertex resolution in the $r-\phi$ plane as a function of the χ^2 cut ($A \rightarrow \tau\tau$ events).*

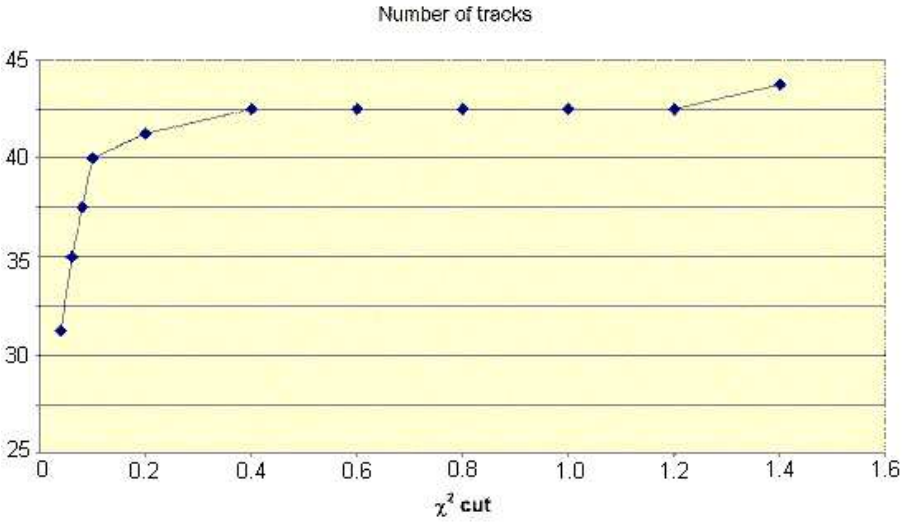


Figure 3.10: *Number of tracks used in the primary vertex fit ($A \rightarrow \tau\tau$ events).*

RMS (cm)		
x	y	z
0.40	0.40	0.51

Table 3.3: *τ vertex resolution.*

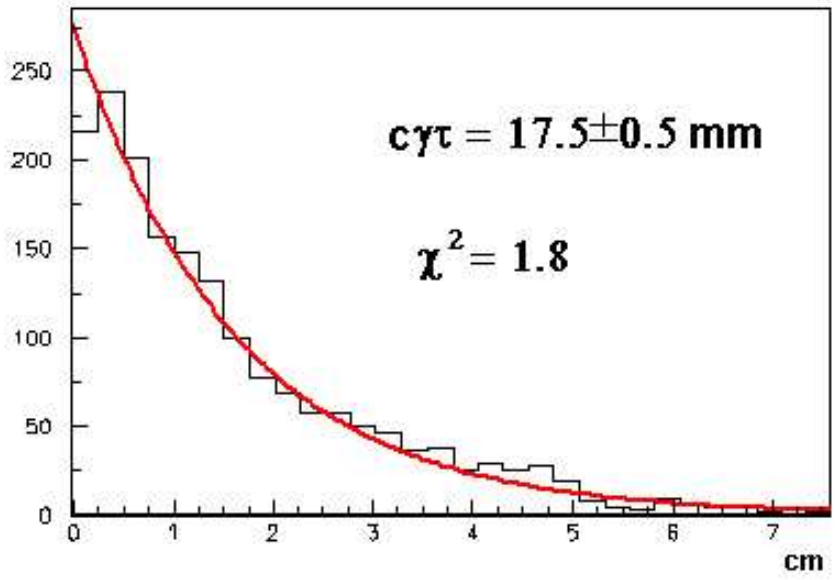


Figure 3.11: *$\gamma c\tau$ for τ's (the vertexing resolution is not included in the error estimation).*

with the the theoretical expectation ($\gamma c\tau \simeq 20\ mm$, for 400 GeV/c average transverse τ momentum).

3.6 Conclusions

An object oriented interface to a vertexing FORTRAN routine has been implemented into the ATLAS offline software. Its capabilities of reconstructing primary and secondary vertices have been tested in $H \rightarrow b\bar{b}$ and $A \rightarrow \tau\tau$ data samples. The results obtained in the test algorithms are encouraging.

Bibliography

[1] R.Cliff, A.Poppelton
IPATREC: inner detector pattern-recognition and track-fitting
ATL-SOFT-94-009

[2] I.Gavrilenko
Description of Global Pattern Recognition Program (XKalman)
ATL-INDET-97-165

[3] P.Billoir
Nucl. Instrum. Methods 225(1984)352

[4] Web page: http://tarta.home.cern.ch/tarta/vtx/docu.html

[5] J.Marriner
Secondary vertex fit with (optional) mass constrains and (optional) pointing constraints
CDF internal note 4/3/1993

Chapter 4

Study of the $A \rightarrow \tau\tau \rightarrow hadrons$ channel

In the MSSM, the $H/A \rightarrow \tau\tau$ rates are strongly enhanced with respect to the Standard Model Higgs decay into two τ's. For low $tan\beta$ values the $H/A \rightarrow \tau\tau$ rate is larger than that obtained for a SM Higgs of the same mass. The $H/A \rightarrow \tau\tau$ branching ratio is about 10% for a mass larger than about 150 GeV/c^2.

The A boson can be produced at LHC via two mechanisms:

- direct production: $gg \rightarrow A$ (see fig. 4.1);

- associated production: $gg \rightarrow b\bar{b}\ A$ (see fig. 4.2).

For low $tan\beta$ values the direct production rates are dominant, while for large values of $tan\beta$ the rates are dominated by the associated $b\bar{b}H$ and $b\bar{b}A$ production.
The relative contribution from the associated production is about 5% for $tan\beta = 5$ and 90% for $tan\beta = 20$, at a mass higher than 500 GeV/c^2.

The two τ decay channel yields the following final state topologies:

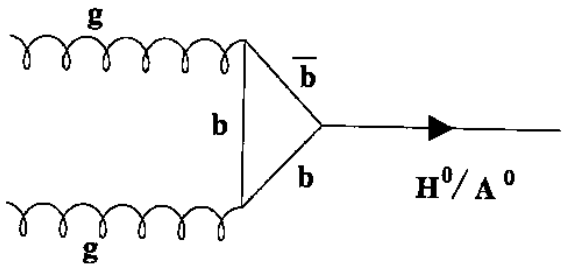


Figure 4.1: *Direct A production.*

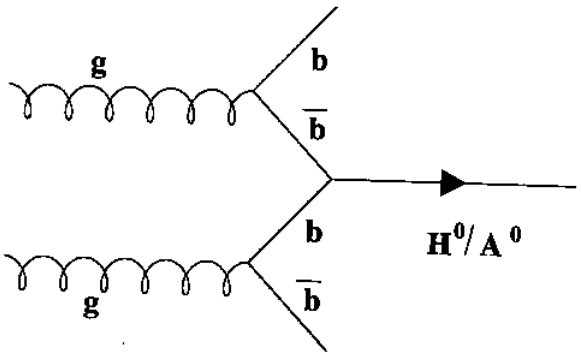


Figure 4.2: *Associated A production.*

- lepton-lepton channel, one τ decays like $\tau \rightarrow (e + \nu_e) + \nu_\tau$ and the other like $\tau \rightarrow (\mu + \nu_\mu) + \nu_\tau$, with a 6.3% branching ratio;

- lepton-hadron channel, one τ decays like $\tau \rightarrow (e/\mu + \nu) + \nu_\tau$ and the other like $\tau \rightarrow hadrons + \nu_\tau$, with a 46% branching ratio;

- hadron-hadron channel (both τ's decay like $\tau \rightarrow \nu_\tau + hadrons$) with a 41% branching ratio.

Only the last topology has been analyzed in this work; the other two channels have been studied in [1], [2], [3], [4] (ATLAS experiment) and in [5] and [6] (CMS experiment).
The analysis performed has to cope with a very huge background, mainly consisting of $t\bar{t}$ and QCD jets: efficient τ tagging and strict selection criteria must be applied.

The working point has been chosen at $m_A = 800$ GeV/c^2 and $tan\beta = 40$. With such a choice of m_A and $tan\beta$ values, only the $b\bar{b}A$ production mode has been studied.

4.1 Cross-section

The associated production ($gg \rightarrow b\bar{b}A$) cross-section has been calculated using the program HQQ [18], while for the total decay widths and branching ratios the HDECAY [7] program has been used. The cross-section times $BR(A \rightarrow \tau\tau)$ used to generate the signal events is shown in table 4.1 (the values used here are the ones reported in [18]).

$q\bar{q}, gg \rightarrow b\bar{b}A \rightarrow b\bar{b}\tau\tau$ (τ's into hadrons)			
m_A (GeV/c^2)	$tan\beta$	σ x BR (fb)	Events
800	40	21.4	25000

Table 4.1: *Associated production cross-section and branching ratio into two τ's, which decay into hadrons.*

4.2 Event production

The signal and background events have been generated with PYTHIA 6.203 [8]. The associated production event topology is sketched in fig. 4.3: two b-jets are associated to two hadronic τ jets.

The generated background events are the most relevant for this kind of signal event topology.

The following backgrounds have been generated:

- **reducible background**

- QCD di-jets (with three different cut on minimum p_T: 50, 100 and 200 GeV/c): this background has a very large cross-section and very high p_T jets which can be misidentified as τ hadronic decays.

- **irreducible background**

- $t\bar{t}$: the t-quark decays into Wb and the W into $\tau\nu_\tau$. The signature is the same as for the signal [1].

- W+jet: the W decays into $\tau\nu_\tau$ (a 10 GeV/c p_T cut has been applied).

- Z+jet: the Z decays into $\tau\tau$.

All the final state τ's undergo hadronic decays. In table 4.2 the background event generation summary is reported.

The data, generated at particle level by PYTHIA 6.203, have been fully simulated using ATLSIM [9], which is a framework to run the ATLAS detector simulations using GEANT3 [10].
A particle level filter has been applied with the following cuts:

- the events are simulated only in the range $|\eta| < 2.7$ (corresponding to the Inner Detector η acceptance);

- at least one parton with $p_T > 17$ GeV/c must be present.

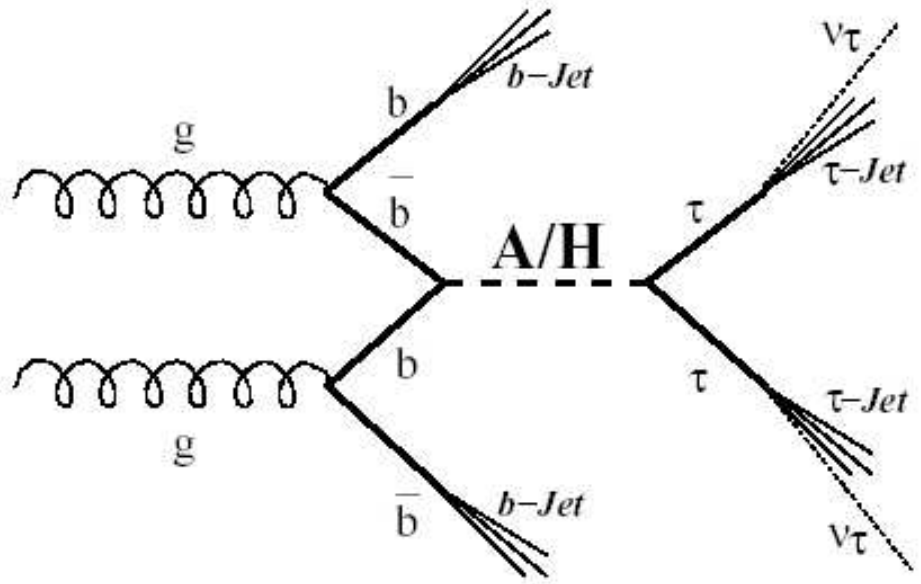


Figure 4.3: *Associated production event topology.*

The filter acceptance/rejection is shown in table 4.3.

A comparison between signal and background cross-section is shown in fig. 4.4 (before and after the filter).

The Pythia generation has been done at CERN, while the full simulation has been mainly performed at the CNAF computing farm in Bologna (about 50000 events have also been simulated in Genoa).
The overall production used about 25000 CPU hours and the data storage is of about 800 Gbytes disk space. The CPU time to simulate an event is reported in table 4.4.
The data files have been permanently stored under the CERN *CASTOR*[11] storage system.

Although a great computing effort has been done to produce the data, these will not be enough to estimate a significative signal/background ratio of the final analysis. Only the signal identification efficiency and a lower limit on the jet-rejection will be calculated. Moreover a satisfying test of the τ-tagging algorithms will be performed.

[1] $t\bar{t} \to hh$ has been ignored as negligible with respect to $t\bar{t} \to \tau\tau$

Channel	Events	p_T cut (GeV/c)	σ x BR (fb)
QCD 50	50000	50	$2.3 \cdot 10^{10}$
QCD 100	50000	100	$1.3 \cdot 10^{9}$
QCD 200	50000	200	$6.3 \cdot 10^{7}$
$t\bar{t}$	100000	-	$2.2 \cdot 10^{5}$
W+jet	100000	10	$6.7 \cdot 10^{7}$
Z+jet	100000	-	$3.6 \cdot 10^{6}$

Table 4.2: *Background generation summary.*

Channel	Generated events	Filtered events	Acceptance	Rejection
Signal	25000	24125	96.5 %	-
QCD 50	50000	40910	-	1.222
QCD 100	50000	48348	-	1.034
QCD 200	50000	49950	-	1.001
$t\bar{t}$	100000	99602	-	1.004
W+jet	100000	53507	-	1.868
Z+jet	100000	58794	-	1.701

Table 4.3: *Data filtering.*

4.3 The reconstruction

The event reconstruction has been done using the Athena framework [12]. A complete reconstruction inside the Inner Detector (ID) and the calorimeters has been performed. The pattern recognition algorithm *xKalman* (see previous chapter) has been used to reconstruct tracks inside the ID. The primary and secondary vertex spotting has been done using the Vertexing algorithms described in the previous chapter. We have used the *tauRec*[19] program both as a mere jet-finding algorithm and to get the τ candidate parameters useful for the τ identification. *tauRec* jet-reconstruction is based on calorimetric seeds and uses a *sliding-window*[13] algorithm with $\Delta\eta\Delta\phi = 0.5$ x 0.5.

Data have also been reconstructed with ATLFAST [14], in order to allow the analysis to access the fast simulation information.

4.4 τ tagging

Since the QCD background cross-section is several orders of magnitude higher than the signal, the analysis must provide a very efficient τ identification and a

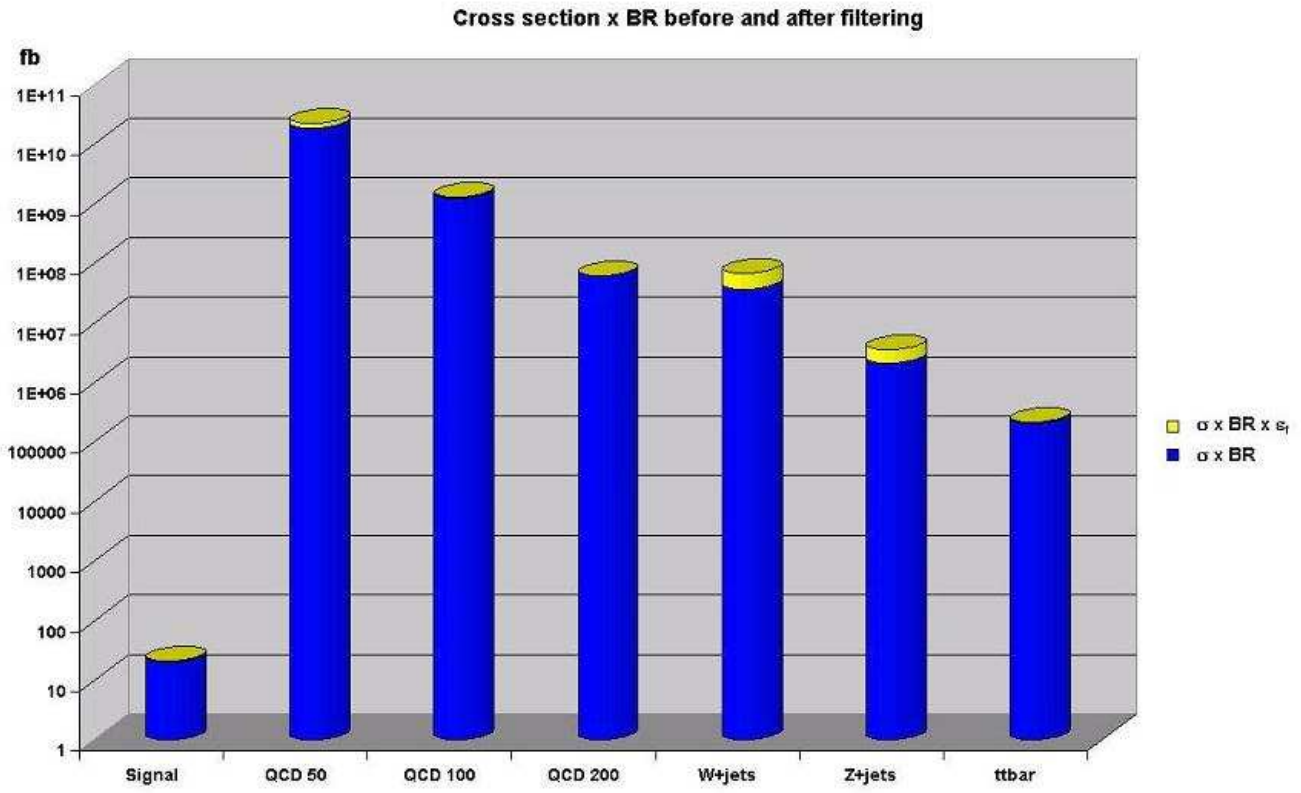


Figure 4.4: *Generation cross-sections before and after applying the simulation filter (entire column: before the filter; dark column: after the filter).*

very good jet rejection.

τ identification has been developed using the fully simulated events in the studies published in [4], in [18] and [15], where mainly the information coming from the calorimeters have been used and where only the *1-prong* decays were selected. This analysis aims to show that the τ identification of the *3-prong* τ-decays can be done more performant exploiting the Inner Detector precise track measurements. Since in this study we look for hadronic τ decays (table 4.5), all the reconstructed jets are considered as candidates to the analysis. These are pre-selected using the high efficiency jet finding algorithm cited in the previous section.

τ-jets are characterized by the presence of a secondary vertex separated from the primary interaction vertex and by tracks in a narrow isolated cone. In a *3-prong* τ-decay, the vertexing algorithm can fit the candidate τ-decay tracks, reconstructing the decay vertex. The existence of a secondary vertex, its distance

Data	*CPU time (s/event)*
Signal	410
QCD 50	140
QCD 100	200
QCD 200	300
$t\bar{t}$	400
W+jet	140
Z+jet	120

Table 4.4: *CPU time to simulate an event.*

from the primary one and the number of tracks used in the fit can be used as discriminating variables in tagging the τ's.
In order to minimize the contamination from particles not coming from the τ decay, the secondary vertex fit is performed using the (at most) three tracks with the highest p_T inside the jet.

Hadronic τ decay modes

Mode	*BR*
1-prong	48.7%
3-prong	15.2%
5-prong	$1.0 \cdot 10^{-3}$

Table 4.5: *Hadronic τ decay modes and branching ratios[17].*

To select *1-prong* τ's, it is required

- $R_{em} < 0.12$, where R_{em} is the jet radius computed in the EM calorimeter using only the electro-magnetic cells contained in the jet;
- one isolated track with $p_T > 40$ GeV/c within a cone of opening[2] $\Delta R < 0.1$ from the jet axis (as evaluated by *tauRec*) and no other track with $p_T > 1$ GeV/c in a cone of $\Delta R < 0.4$.

3-prong τ's are pre-selected requiring

- *1-prong* selection failed;
- $R_{em} < 0.12$;

[2] $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$

- one isolated track with $p_T > 40$ GeV/c within $\Delta R < 0.1$ from the jet axis (as leading track), at most two other tracks with $p_T > 1$ GeV/c in a cone of $\Delta R = 0.03$ around the leading track and no other track (*isolation cut*) with $p_T > 1$ GeV/c in an annulus of $0.03 < \Delta R < 0.4$ (see fig. 4.5);

The further *3-prong* τ-decay selection requires a successful decay vertex fitting.
3-prong τ candidates are eventually selected if they satisfy one of the following requirement sets (used in *OR*):

- **Set 1**
  - the χ^2 contribution of the tracks fitted to the secondary vertex smaller than the contribution of the same tracks to the primary;
  - the number of the tracks fitted to the vertex equal to 3;
  - $p_{T3} > 7$ GeV/c (with p_{T3} the transverse momentum of the third most energetic track).
- **Set 2**
  - 3 tracks fitted to the vertex;
  - $p_{T2} > 40$ GeV/c (with p_{T2} the transverse momentum of the second most energetic track).
- **Set 3**
  - $d_{r\phi}^{12} > 0.4$ cm (with $d_{r\phi}^{12}$ the distance, projected onto the $r\phi$ plane, between the primary and secondary vertex;
  - 3 tracks fitted to the vertex;
  - $p_{T1} > 50$ GeV/c (with p_{T1} the transverse momentum of the most energetic track);
  - $p_{T2} > 25$ GeV/c.
- **Set 4**
  - $d_{r\phi}^{12} > 0.4$ cm;
  - 2 tracks fitted to the vertex;
  - $p_{T1} > 50$ GeV/c;
  - $p_{T2} > 35$ GeV/c.

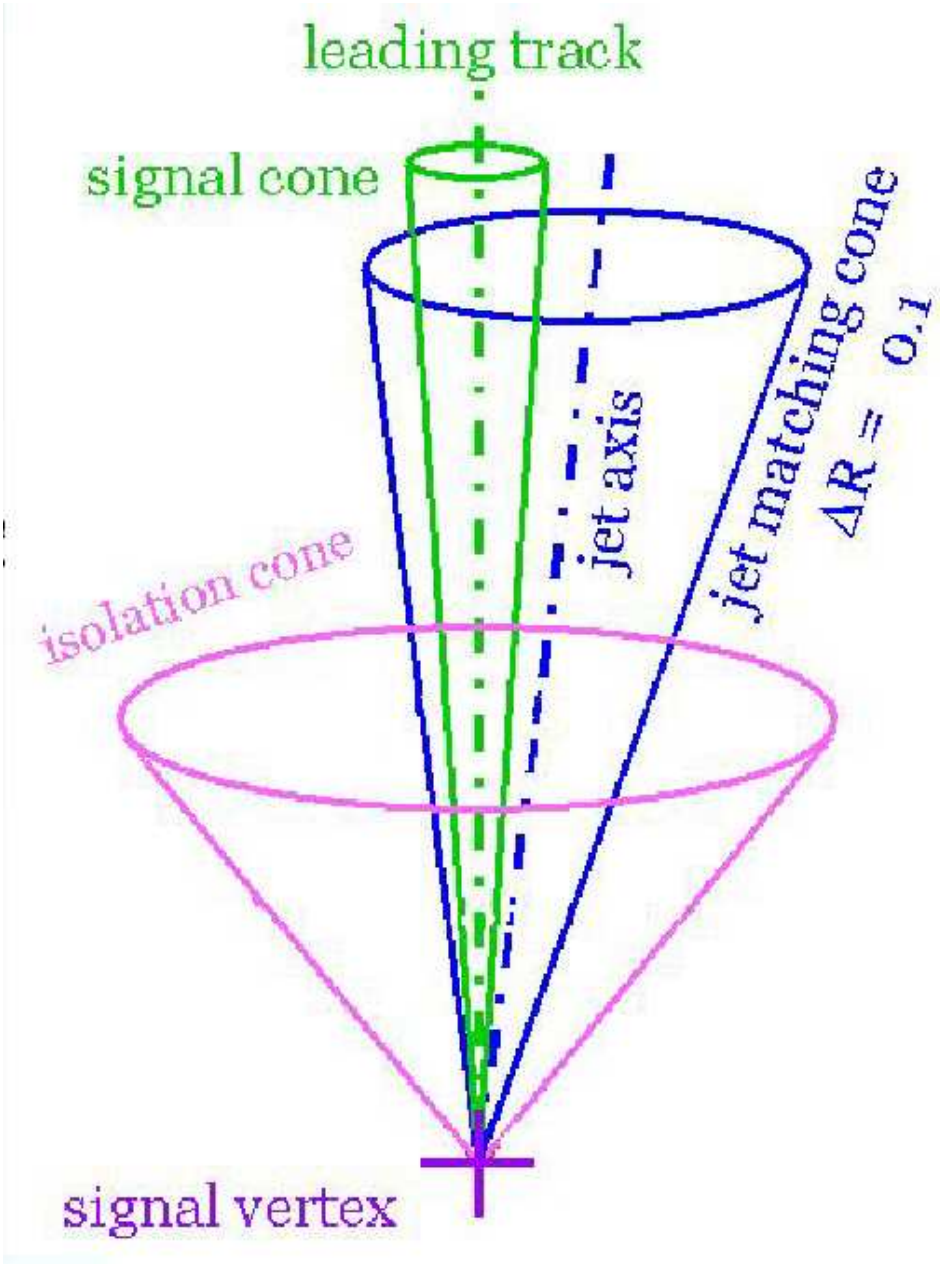


Figure 4.5: *Isolation criteria.*

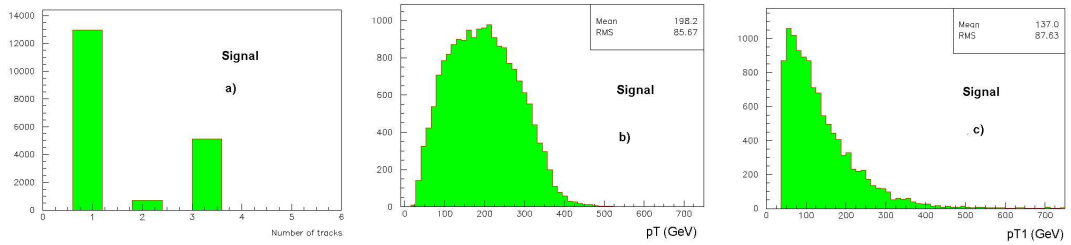


Figure 4.6: *Signal: a) number of tracks to be fitted, b) p_T and c) p_T of the most energetic track.*

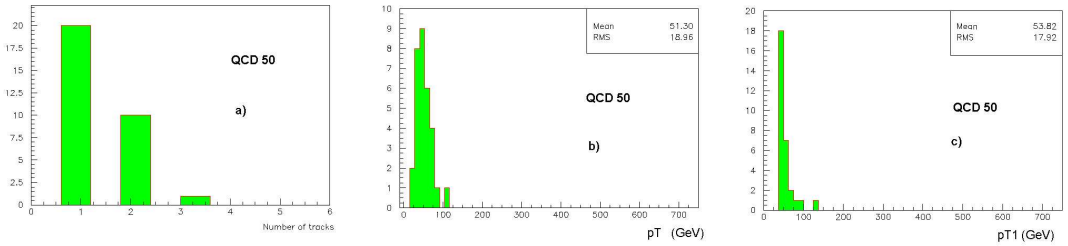


Figure 4.7: *QCD50: a) number of tracks to be fitted, b) p_T and c) p_T of the most energetic track.*

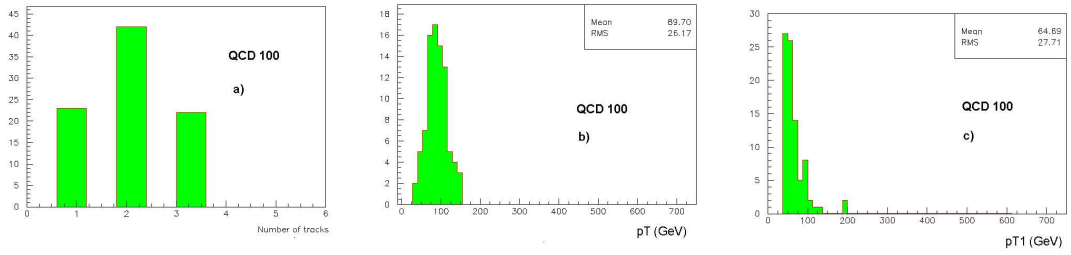


Figure 4.8: *QCD100: a) number of tracks to be fitted, b) p_T and c) p_T of the most energetic track.*

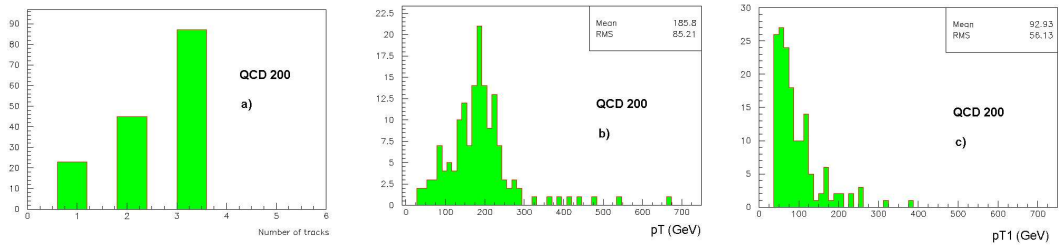


Figure 4.9: *QCD200: a) number of tracks to be fitted, b) p_T and c) p_T of the most energetic track.*

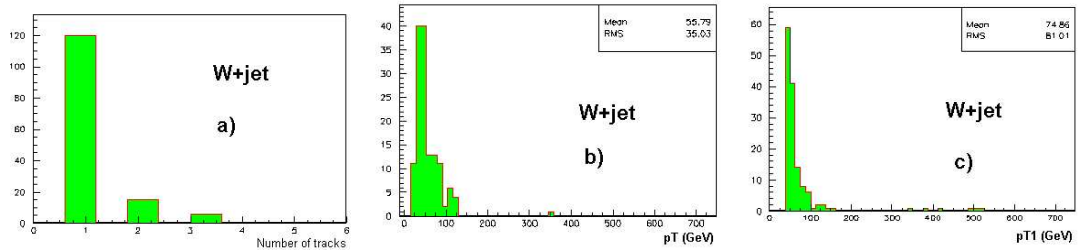


Figure 4.10: *W+jet: a) number of tracks to be fitted, b) p_T and c) p_T of the most energetic track.*

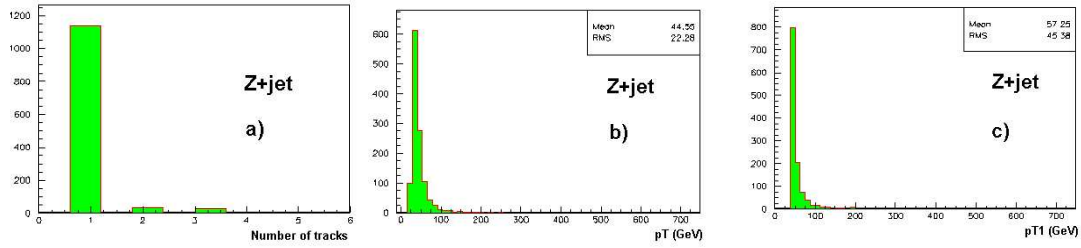


Figure 4.11: *Z+jet: a) number of tracks to be fitted, b) p_T and c) p_T of the most energetic track.*

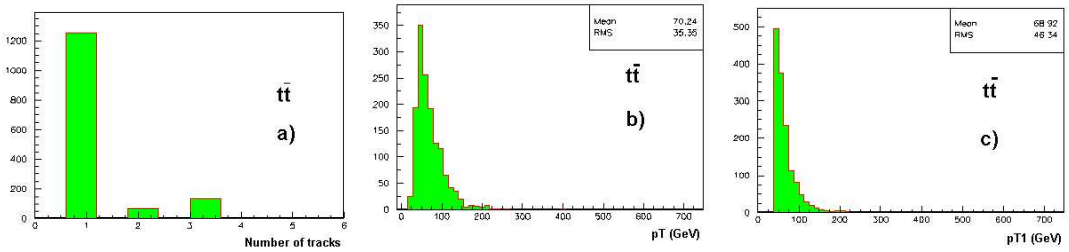


Figure 4.12: *$t\bar{t}$: a) number of tracks to be fitted, b) p_T and c) p_T of the most energetic track.*

The cuts have been optimized in order to achieve a high rejection rate on the background.

In figs. 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12 the number of tracks to be fitted, the τ candidate p_T and the most energetic track p_T are respectively shown for signal and background (the τ candidates are those selected after the isolation cut).

The same τ-tagging selection has been applied both on signal and background filtered events; the results are summarized in table 4.6. Single decay mode efficiency has also been calculated: *1-prong* efficiency is 34.4 % and *3-prong* one is 44.1 %, with a 10 % enrichment with respect to the previous analysis in [18].

Data sample	*Filtered τ/jets*	*1-prong*	*3-prong*	*tagged τ's*	*Efficiency*	*jet rejection*
Signal	48250	12932	4704	17636	36.6%	-
QCD50	81820	20	$< 2.4^{\dagger}$	$< 22.4^{\dagger}$	-	$> 3.7 \cdot 10^3$
QCD100	96696	22	10	32	-	$3.0 \cdot 10^3$
QCD200	99900	28	71	99	-	$1.0 \cdot 10^3$

Table 4.6: *τ-tagging results for signal and QCD background. † Poissonian upper limits for a 90% confidence interval [17].*

As expected, a part of the τ's coming from t-quark, W and Z decays has been tagged (see table 4.7).

No trigger pre-selection has been applied, because of the difficulties of reproducing an on-line cut at this stage. One of the proposed ATLAS τ-triggers requires $E_T^{\tau} > 20$ GeV and $E_T^{miss} > 30$ GeV (with E_T^{τ} the τ's transverse energy and with E_T^{miss} the transverse missing energy). Although a small correlation

Data sample	*Filtered τ/jets*	*1-prong*	*3-prong*	*tagged τ's*
$t\bar{t}$	199204	1250	82	1332
W+jet	53507	120	3	123
Z+jet	117588	1137	21	1148

Table 4.7: *τ-tagging results for irreducible background.*

between trigger-level and reconstructed jet energy has been found [4], a 30 GeV cut on E_T^{miss} has also been applied. The E_T^{miss} calculation requires a very good knowledge of the calorimeters calibrations. Such studies have not yet been completed in a full simulated scenario using the new software; this is the reason why the E_T^{miss} has been calculated using the program ATLFAST [14].

4.5 $A \rightarrow \tau\tau$ analysis

The described τ-tagging algorithm is applied in order to select events with two τ's, as candidates for the A boson identification.

The analysis (similar to the one reported in [3]) can be described as follows:

1- only the events where two jets have been τ-tagged (as described in the previous section) are kept. This requirement completely remove all the QCD an W+jet events;

2- no electron or muon with $p_T > 10$ GeV/c is allowed, since we are looking for fully hadronic events;

3- the jets with $p_T > 20$ GeV/c and $|\eta| < 3.2$ have not to be more than 4 (to suppress $t\bar{t}$ background);

4- since the A boson is produced in association with two b-quarks, at least one b-tagged jet is required (the ATLFAST output is used to find b-tagged jets, assuming a 70% b-tagging efficiency);

5- high missing energy ($E_T^{miss} > 65$ GeV) is required to select events where high energy neutrinos have been produced;

6- the $\Delta\phi$ angle between the two τ's has to be in the range $145° < \Delta\phi < 172°$ to reject QCD events, where two back-to-back jets are expected. The Z+jet channel is also suppressed by this cut since the Z and the jet are mainly produced back-to-back and the τ's decayed from the Z usually are also back-to-back;

7- in order to suppress events containing W's, the transverse mass of the $\tau - p_T^{miss}$ system is calculated for both τ's. The transverse mass is defined as

$$m_T = \sqrt{2p_T^{\tau} p_T^{miss}(1 - cos(\Delta\phi(\tau, p_T^{miss})))}$$

where $\Delta\phi(\tau, p_T^{miss})$ is the ϕ angle between the τ-jet and the p_T^{miss} directions. The minimum of $m_T(\tau_1, p_T^{miss})$ and $m_T(\tau_2, p_T^{miss})$ is used in the selection. High m_T values are expected in events containing W's, while smaller ones are expected in the signal. The cut set is $m_T < 50$ GeV/c^2;

8- The final cut is on the reconstructed m_A mass [18]. It can be written as

$$m_A = \sqrt{2(E_1 + E_{\nu_1})(E_2 + E_{\nu_2})(1 - cos\theta)}$$

where E_i is the i-th τ energy, E_{ν_i} is the energy of the i-th neutrino system and θ is the angle between the two τ directions. This formula is valid under the approximation that $m_\tau = 0$, that the directions of the two neutrino systems from each τ-decay are coincident with the directions of the measured τ-decay products and finally under the condition that the τ-decay products are not back-to-back.
E_{ν_1} and E_{ν_2} are found solving the following system:

$$\begin{cases} E_{\nu_1} sin\theta_1 cos\phi_1 + E_{\nu_2} sin\theta_2 cos\phi_2 = P_x^{miss} \\ E_{\nu_1} sin\theta_1 sin\phi_1 + E_{\nu_2} sin\theta_2 sin\phi_2 = P_y^{miss} \end{cases}$$

where θ_i and ϕ_i describe the i-th τ direction. The system cannot be solved if the determinant vanishes, which happens if $sin\Delta\phi_{\tau_1,\tau_2} = 0$ (with $\Delta\phi_{\tau_1,\tau_2}$ the $\Delta\phi$ separation between the two τ's). Moreover the reconstructed m_A mass depends on $\Delta\phi_{\tau_1,\tau_2}$ (see fig. 4.13): if $\Delta\phi_{\tau_1,\tau_2}$ is close to π the neutrino system yields bad solutions which generate tails at high values of the reconstructed mass. It may also happen that the accuracy in the measurement of p_x^{miss} and p_y^{miss} in the calorimeter and the assumptions made result in unphysical negative solutions for E_{ν_1} and E_{ν_2}: no m_A can be reconstructed in this case. The m_A distribution undergoes a gaussian fit and only the value within a 1.5σ range around the mean value are kept.

In fig. 4.14 the distributions of the variables corresponding to cut 5 to 8 are shown.

In table 4.8 the signal efficiency and the number of surviving background events are reported.

The $t\bar{t}$ background is the most difficult to cope with in this analysis. No background event survives the final cut but, as expected (see section 4.2), the analysis is affected by insufficient statistics which prevents us to be conclusive.

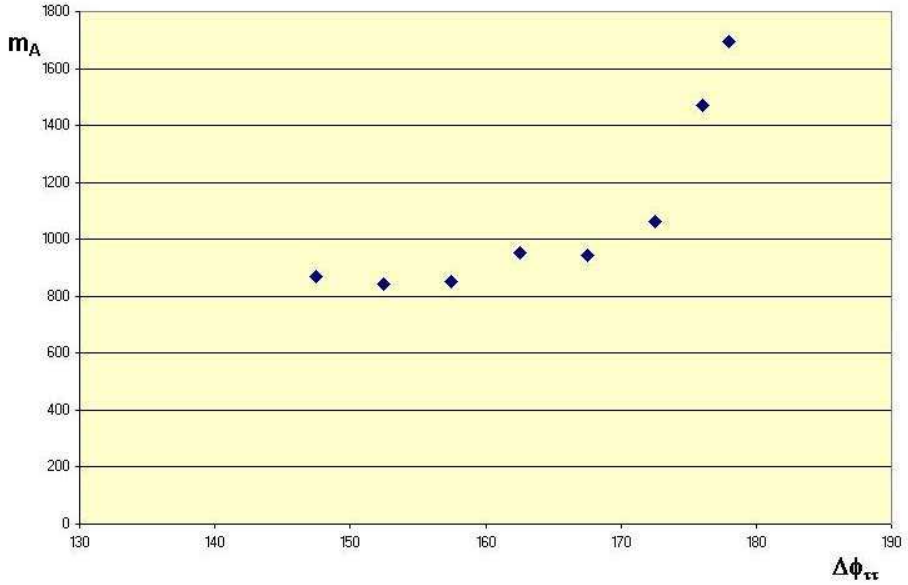


Figure 4.13: *Reconstructed m_A mass (for a 800 GeV/c^2 A Higgs boson) versus the $\Delta\phi$ separation between the two τ's.*

4.6 Conclusions

A τ identification algorithm, which mainly uses the ATLAS Inner Detector measurements, has been introduced. The effort has been focused on 3-prong τ decays: isolation criteria and the Vertexing tools described in the previous section have been used to identify τ's. This channel contribution to the overall τ-tagging efficiency has been around 10 %.

An analysis to study the channel $A \rightarrow \tau\tau \rightarrow hadrons$ has been developed and tested. The τ-tagging has been used in this analysis as a fundamental tool to preselect A Higgs candidates.

A large amount of full simulated data has been produced exploiting all the computing power available.

Since the background cross-sections are several orders of magnitude higher than the signal, severe criteria have been applied in the analysis. No background events survive the last cut and a 0.5 % signal efficiency has been achieved. Although the results are promising, an insufficient statistics prevents any further conclusion.

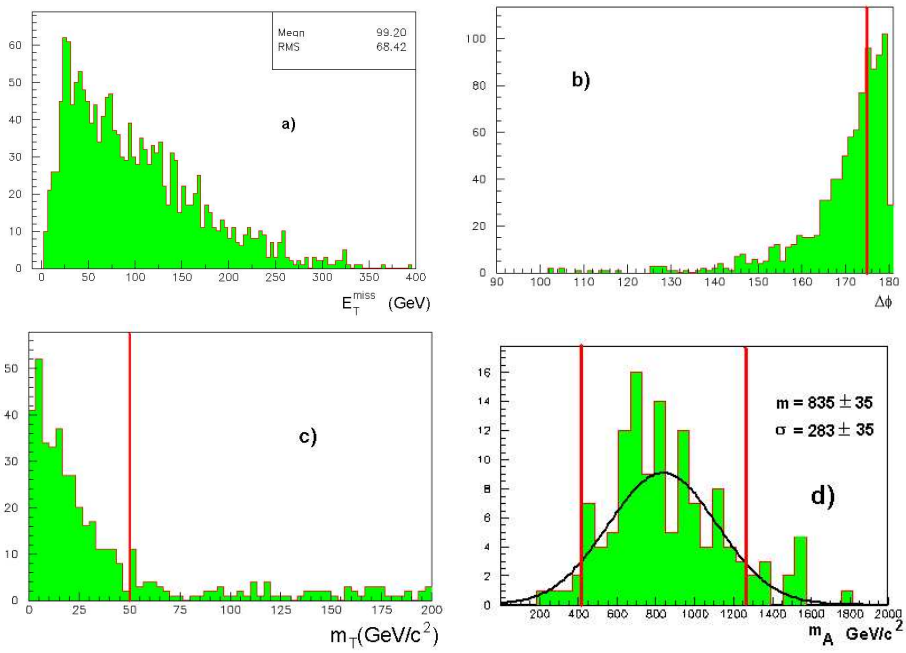


Figure 4.14: *a)* E_T^{miss} *distribution on the signal after the 4-th cut. b)* $\Delta\phi$ *between* τ *candidates after the 5-th cut. c) Transverse mass* m_T *in signal events, after the 6-th selection cut. d)* m_A *distribution on the signal: gaussian fit performed.*

Cut	Signal	Efficiency	$t\bar{t}$	Z+jet	*Total bkg.*
1	3916	16.2 %	5	14	*19*
2	3558	14.7 %	5	14	*19*
3	3503	14.5 %	5	14	*19*
4	1632	6.8 %	5	0	*5*
5	1000	4.1 %	4	0	*4*
6	585	2.4 %	4	0	*4*
7	245	1.0 %	1	0	*1*
8	110	0.5 %	0	0	*0*

Table 4.8: *Number of selected events.*

Bibliography

[1] D. Cavalli, L. Cozzi, L. Perini, S. Resconi
ATL-PHYS-1994-051 (1994).

[2] D. Cavalli, G. Negri
ATLAS Note in preparation (2002).

[3] J.Thomas
Study of heavy MSSM-Higgs-Bosons A/H in hadronic τ-decays in ATLAS
ATLAS note in preparation.

[4] ATLAS Coll.
ATLAS Detector and Physics Performance: Technical Design Report, report CERN/LHCC 99/14-15 (1999).

[5] Sami Lehti
Dissertation, University of Helsinki, Report Series in Physics, HU-P-D93, 2001.

[6] R. Kinnunen and A. Nikitenko
CMS NOTE 1997/106.

[7] A. Djouadi, J. Kalinowski and M. Spira, Comp. Phys. Comm. 108 (1998) 56.

[8] http://www.thep.lu.se/ torbjorn/Pythia.html

[9] http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/DOCUMENTS/ATLSIM/Manual/main_manual.html

[10] http://wwwinfo.cern.ch/asd/geant/index.html

[11] http://castor.web.cern.ch/castor/

[12] http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/architecture/General/index.html

[13] R.Mehdiyev et al., *Test of the Sliding Window Algorithm for Jets Reconstruction in ATLAS Hadronic Calorimeters*
ATL-CAL-99-002

[14] E.Richter-Was et al.
ATLFAST 2.0 a fast simulation package for ATLAS
ATL-PHYS-98-131

[15] D.Cavalli et al.
Combined analysis of $A \rightarrow \tau\tau$ events from direct and associated bbA production
ATL-PHYS-2000-005

[16] R.Kinnunen, D.Denegri
The $H_{SUSY} \rightarrow h^{\pm} + h^{\pm} + X$ channel, its advantages and potential instrumental drawbacks
CMS note 1999/037

[17] Particle Data Group
2002 Review of Particle Physics
Physical Review D66, 010001 2002

[18] D.Cavalli et al.
Study of the MSSM $A/H \rightarrow \tau\tau$ at the LHC
included in: *The Higgs Working Group Summary Report*
hep-ph/0203056 - 5 Mar 2002

[19] H.Ma, F.E.Paige, S.Rajagopalan
τ Reconstruction in Athena
ATLAS note in preparation

Chapter 5

Summary

ATLAS is one of the five experiments which will take data at the *Large Hadron Collider* in construction at the CERN. The accelerator start is scheduled for the year 2007 and all the experiments are, at the time being, under preparation.

This challenging project aims mainly to probe the Standard Model (SM) and the physics beyond. Even though the SM has been verified at a high precision level by the LEP and SLD experiments, the problem of the Higgs mechanism nature and that of the mass hierarchy are still open, together with the general framework beyond the SM.

The most likely candidates to play this role seem to be the supersymmetric (SUSY) theories, whose main ingredient is the introduction of a symmetry between bosons and fermions. SUSY solves many of the open SM questions, but need to introduce a large amount of new particles never seen before: each SM boson/fermion will have a fermionic/bosonic partner.
The MSSM is the minimal supersymmetric extension of the SM. In its framework each SM particle has its supersymmetric partner and the single SM Higgs is replaced by five physical Higgs field states (h, H, A, H^+, H^-).

In order to have enough energy to generate these alleged new particles, the LHC will accelerate two proton beams of 7 TeV each. The events produced will be very complicated and the interesting ones will be submerged by an overwhelming QCD background. The detector of the LHC experiments have been designed to face this high multiplicity scenario. In particular, the sub-detectors near the interaction region have been built with a very high granularity in order to allow an efficient tracking and vertexing.

Together with the detector construction, a big effort is being done by the LHC collaborations to develop the necessary software architecture. It will manage the detector measurements, will create the simulations and will analyze the data taken in the runs.

In this thesis a package aimed to reconstruct primary and secondary vertices has been developed. It is an Object Oriented interface to an existing vertexing routine written in Fortran by the CDF experiment. The package, which has

been officially released in the ATLAS software, has been tested and has given encouraging results.

This tool has been widely used to develop an algorithm aimed to identify the hadronic τ decays. In particular, it has been useful in spotting the decay vertices of τ's decaying into three prongs. This τ-tagging algorithm has been used to identify τ's in the $A \rightarrow \tau\tau \rightarrow hadrons$ channel analysis.

This analysis has been developed following the guidelines of already published works. About 450000 background and 25000 full simulated events have been produced and analysed in this thesis, requiring an outstanding computing effort. Since the background cross-section is several orders of magnitude higher than the signal, the analysis criteria have been focused to a strong background suppression. The τ-tagging, which can be considered a pre-selection step, has been proved as an affordable and useful tool.

A 0.5% efficiency and no background events left are the final analysis numbers. The results are considered encouraging but cannot be definitive, since much more background should be produced in order to achieve the necessary statistics. At least 10^7 background events should be generated to test significantly the analysis developed, which would require much more computing power and data storage.

Appendix A

Appendix

This chapter is intended to provide a detailed description of the vertexing packages of the ATLAS Offline Software. The code description is addressed to the readers who want to go deeper inside the software technicalities and who have a basic knowledge of the ATLAS software structure.

A.1 Introduction

A set of packages which perform the reconstruction of primary and secondary vertices has been provided.
The packages are stored in the offline software *repository*, which is the container of the all ATLAS software. The code stored inside the *repository* is maintained by mean of a system for the version control (*CVS*) which allows the developers to keep a history of the code modifications. Each file version is made distinguishable by mean of a label (tag). The most recent and last tagged code is periodically compiled and published in a software *release*. Each release is a self-consistent copy of the most up to date working ATLAS software. The *Vertexing* package is inside the official release since release tagged as 2.4.0. It has been implemented as a sub-package of the *Reconstruction* container package.

The code is written with an Object Oriented strategy, using the C++ language. The algorithms have been created to work properly in the official framework used in the ATLAS Reconstruction: Athena [1].
The vertex fit is performed by a program developed by the CDF collaboration and adapted to the ATLAS standards. Since the source code is FORTRAN, it has been included in the *External* container package and an interface to wrap it into C++ has been implemented.
These packages allow the user to take the track objects that come from the pattern recognition, to fit them to a vertex and to write out the reconstructed vertex object.
The *Vertexing* package has the following sub-packages structure:

- *VertexFit*: the mere interface.

- *SimpleVertex*: the object containing the vertex information.

- *VertexPrimary*: the primary vertex fitting algorithm.

- *VertexExampleSecSeed*: an example about how to reconstruct secondary vertices starting from a seed.

- *VertexExampleSecUserTracks*: an example about how to reconstruct secondary vertices using a user-defined list of tracks.

- *VertexExampleScheme*: a dummy example which summarizes the examples code structure.

A.2 The FORTRAN code: *CTVMFT*

The main fitting routine in *CTVMFT* was written for the CDF experiment Run I and its I/O was later adapted to the ATLAS standards [2]. The I/O procedure is performed using FORTRAN common blocks (see details in figs. A.1, A.2 and A.3) and several subroutines are called during the execution of the program main stream.
The scheme of the common blocks and routines can be seen in fig. A.4. The code in the subroutine *gettrk* has been rewritten to enable the program to read the tracks parameters directly from a common block filled by the interface, the original code taking these parameters from a ZEBRA bank, which is not created any more by the pattern recognition algorithms.

A new object (*SimpleTrack*) stores the tracks information. The tracks parameters are read by mean of the interface (see section A.4), which accesses them in the *Transient Data Store* (TDS) and saves them to memory inside common block shaped C structures.

The reconstructed vertex position is calculated using a χ^2 minimization method. The resulting position minimizes the sum of the single track χ^2 contributions to the vertex.
The minimization routine starts from the intersection point of the first two tracks to fit. The distance between this point and all the tracks as well as the derivatives matrix are computed and they are used as the minimization routine input. If the fit converges, the results are copied in the output common blocks.

The FORTRAN routines scheme is the following:

- *CTVM00*: checks the input for completeness and consistency; if an error is detected in the input conditions, it is considered as fatal and the fit is

aborted. For instance, an error occurs if the primary vertex covariance matrix is found to be singular or if no tracks are associated to the vertex.

- *CTVMFA*: makes the vertex first approximation. The two helices corresponding to the first two tracks specified in the calling sequence are projected onto the r, ϕ plane. If the two circles intersect, the point with smaller z difference is selected as the first estimated vertex position; if the circles do not intersect, the vertex approximation is taken as the point of closest approach of the two circles.

- *CTVMVF*: calculates the contributions from each track (at the vertex) to the derivative matrix to be used in the fit.

- *CTVMPF*: if any pointing constraint is specified, it calculates the constraints contribution to the derivative matrix for the vertex to be fitted. The pointing constraints constrain the vertex momentum to point to another vertex.

- *CTVMMF*: if any mass constraint is specified, it calculates the constraints contribution to the derivative matrix for the vertex to be fitted.

A.3 The object *SimpleVertex*

The *SimpleVertex* object is derived directly from a *DataObject* base class, like any other object to be put in the TDS.
The *SimpleVertex* object contains a set of variables and methods which allow the user to store and access all the vertex properties. In addition to the main vertex properties, a few lists of *SimpleTrack* pointers are saved into the object. These pointers are:

- the list of the tracks used in the fit;

- the list of the used tracks coming from the primary vertex (allowing the access to the Monte Carlo information);

- the list of the used tracks coming from the secondary vertex fitted (allowing the access to the Monte Carlo information);

- the list of the tracks used in the fit, as changed by the fitting algorithm.

The complete list of the variables is shown in fig. A.5.

These pointer lists need to be explicitly deleted in order to avoid memory leak during the program execution. Since these lists are part of the *SimpleVertex* object and are to be stored in the TDS, they cannot be erased by the program algorithm but need to be deleted by the *SimpleVertex* destructor ($\sim SimpleVertex()$).

A.4 The interface

The package *VertexFit* contains the mere *Vertexing* interface. Its aim is to allow the user's C++ algorithms to communicate with the *CTVMFT* FORTRAN code. The common block read by *GetTrack* is filled using the class *Stio*, by mean of several methods and of a C structure that emulates the FORTRAN block. The track parameters are taken from *SimpleTrack* objects (created by the pattern recognition algorithm, e.g. *xKalman*) and are stored in the structure.
Vf_io class contains the methods to set all the fit parameters and to retrieve the fit results. Two C structures emulate *ctvmfq* and *ctvmfr* blocks, which are responsible of the *CTVMFT* I/O.
The possibility of evaluating the single track contribution to the vertex fit χ^2 is provided as well. The track with the highest χ^2 can be spotted, marked and, if the case, excluded from the new list of the tracks to be fitted. This procedure can be iterated until at least one track exceeds the user's set χ^2 limit. If less than two tracks are left, the fit is aborted.
A new class called *lego* has been implemented to allow this machinery to work. This class allows the interface to establish a link between the C++ *SimpleTrack* pointer and the respective FORTRAN vector index, as well as to store tracks information such as χ^2 contribution to the vertex.
The parameters used to specify the vertex properties are stored in the *Flags_n_Par* class, where the flags and the controllers which are necessary to perform the previous features ae also stored. A list of the *Flags_n_Par* variables can be seen in fig. A.6.

Setting the *FIT_MODE* variable, the user can toggle the three operating modes:

- **mode 1**: only the primary vertex is fitted. Only one vertex per event is looked for;

- **mode 2**: only secondary vertices (if any) are fitted. The primary vertex position and the covariance matrix are *hardwired* (put by hand in the code by the user, if different from defaults);

- **mode 3**: both primary and secondary vertices are fitted. Primary vertex fit results are used as input for the secondary vertex fits. The secondary vertex fits are skipped if no primary vertex is found.

A.5 The algorithms

Four Athena algorithms have been written and included in the *Vertexing* package; three of them are intended as examples, aimed to show the main interface features. The algorithms names are self-explanatory:

- *VertexPrimary:* only the primary vertex is fitted;

- *VertexExampleSecSeed:* both primary and secondary vertices are fitted. The secondary vertex is looked for starting from a seed, such as a calorimetric cluster or a simulated information;

- *VertexExampleSecUserTracks:* both primary and secondary vertices are fitted. The secondary vertex is fitted using a list of tracks provided by the user;

- *VertexExampleScheme:* a dummy algorithm summarizing the common steps of the previous examples.

The algorithms are divided into two parts: a first part in which all the objects, flags and variables are initialized; a second one in which the algorithm methods are called.
In fig. A.7 the first part of *VertexExampleScheme* is shown.
The object *Vt* is a temporary vertex used to store the vertex information before the permanent vertex object is filled.
The object *Vs* is used to handle the simulated information when available.
The vector *SEED* stores the information about all the seeds found.
Flags *M3* and *SECFLAG* control the iteration upon primary and secondary vertices.
Vcoll is a collection of *SimpleVertex* objects reconstructed in the program. This collection is saved in the TDS after reconstructing all the vertices.

In fig. A.8 the methods called in the algorithms are shown.

The vertex seeds are used when the mode is not '1' and there is no user provided track list (such as the case of *VertexExampleSecSeed* algorithm). The method *Vs.GetSeed(...)* returns the number of seeds found and the vector *SEED* that contains all the seed information.
Before starting the fit, the reconstructed tracks collection is retrieved from the TDS: the constant vector *vec* is declared as *DataHandle* and then used by the *StoreGate* retrieve method. If the track container cannot be retrieved, an error message is printed out and the algorithm execution is interrupted.

In order to look for both primary and secondary vertices, all the methods needed to perform the fit are included in a *do* loop which terminates when the *SECFLAG* flag reaches the number of selected seeds; if an external track list is provided, the loop continues until a user set number of vertices has been fitted.
Inside the loop two lists of tracks are created: the list of the user provided tracks (*MyTracks*) and the one of the selected tracks to fit (*SelectedTracks*). These lists are erased at the end of the loop.
The temporary vertex object is cleared with *ClearVt()* and the tracks are selected by *TrackSelection* method.

If the user requires the Monte Carlo information about the vertex, it is retrieved by *TrackSim* method and stored in the temporary vertex.
The fit routine is called by *DoFit*: *FnP* object provides the fit parameters, *Vt* object stores the results and *SelctedTracks* provides the tracks to fit.
After the fit, the vertex object is filled. If the fit fails a dummy vertex is stored as well.
Once the loop is completed, the vector object containing the vertices is saved in the TDS.

A.6 The *jobOptions* files

At the beginning of any algorithm, a set of variables can be declared as algorithm *properties*. Their value can be set by mean of a text file that is read by Athena during the algorithm execution (by default *jobOptions.txt* is read). This feature allows the user to run the program with different parameters without recompiling the code, but simply changing the options file. If a property value is not declared in the options file and no default is specified in the code, a runtime error occurs. As to *Vertexing* algorithms, in fig. A.9, A.10, A.11 the *jobOptions* files of *VertexPrimary*, *VertexExampleSecSeed* and *VertexExampleSecUserTracks* are respectively shown.

The first two properties are needed by the Athena application manager and declare the shared library and the top algorithm name.
OutputLevel property chooses the printout level to be used in the algorithm.
The *GENERAL* properties set the algorithm running mode: the user can decide whether to fit the vertices and what kind of vertex to look for.
In the primary vertex properties, the user can set a few track quality cuts and can decide whether to use the χ^2 loop iteration.
In the secondary vertex properties, the user can choose the track selection parameters, can enable/disable the χ^2 loop and can decide how many mass constraints are to be used (from 0 to 4).

A.7 The dependencies

This package needs to be linked to other parts of code contained in different packages (for instance, an algorithm may need to use a class or a method defined outside the package it belongs).

In order to let the compiler know the packages it needs to link, a so-called *requirement* file is provided. In figs. A.12, A.13, A.14, A.15 and A.16 the *Vertexing* subpackages *requirement* files are shown.

A few packages are always included since they contain the code necessary to

the framework to work properly and some useful utilities and tools (for instance, *StoreGate* is the software which handle the I/O to/from the TDS).

Other packages are included only in some *requirement* files. *SimpleTrack* is needed to handle the tracks reconstructed by the pattern recognition algorithm. *Genz*, *ggenz* and *geant3* are needed in order to retrieve the simulated information. *CTVMFT* is needed by the inrface only, in order to access the fitting routine. *ZebraTDREvenet*, *EventAthena* and *EventStructure* are required to access data in the proper way and with the correct format.

```
Dimension Parameters
INTEGER MaxVtx,MaxTrk,MaxMcn,MaxItr, MAXDIM
PARAMETER (MaxVtx = 3) ! maximum number of vertices
PARAMETER (MaxMcn = 4) ! maximum number of mass constraints
PARAMETER (MaxTrk =10) ! maximum number of tracks
PARAMETER (MaxItr =10) ! maximum number if iteration steps

INTEGER UDIM
PARAMETER ( UDIM = MaxTrk*(MaxVtx+MaxMcn+60)
+ MaxVtx*(MaxVtx+18) + MaxMcn*6 +MaxItr + 31 )

PARAMETER (MAXDIM = 7*(MaxVtx+1) + 3*MaxTrk + MaxMcn)


Required Input Fit Specification
INTEGER NVERTX ! Number of vertices
INTEGER NMASSC ! Number of mass constraints
INTEGER NTRACK ! Number of tracks, this fit

LOGICAL TRKVTX(MaxTrk,MaxVtx) ! (Track,Vertex) association table
INTEGER VTXPNT(MaxVtx,2) ! vertex association information
INTEGER CVTX(MaxVtx) ! Conversion constraint specification
LOGICAL TRKMCN(MaxTrk,MaxMcn) ! (Track,Mass Constraint) table
REAL CMASS (MaxMcn) ! Constraint mass

Required Input Track Specifications
CHARACTER*4 TKBANK(MaxTrk) ! Track bank type (e.g. 'TRKS')
INTEGER LIST (MaxTrk) ! Track bank number
REAL TMASS (MaxTrk) ! Track mass assignment

Added by Leanne Jan 98
Original track parameters and covariance matrix

REAL TRK0 (5,MaxTrk) ! Original track parameter values
REAL COV0 (15,MaxTrk) ! Original track covariances


Fit input, collected using the input specifications
REAL PAR0 (5,MaxTrk) ! Original track parameter values
REAL G (5,5,MaxTrk) ! Fitted track parameter covariances

Optional Input, Required Data for Pointing Constraints
REAL XYZPV0(3), EXYZPV(3,3) ! "Primary" vertex; XYZ, covariance

INTEGER RUNNUM,TRGNUM ! "you can't tell the players without a score card"
INTEGER ITER ! number of iterative steps taken
INTEGER NTSCUT ! total number of cut steps, this fit
```

Figure A.1: *ctvmfq and ctvmfr common blocks (input).*

```
Output; Results of the Fit
INTEGER MATDIM ! Dimension of the matrix, this fit
INTEGER NDOF ! Fit number of degrees of freedom
LOGICAL VtxVtx(MaxVtx,MaxVtx) ! vertex geneology
REAL CHISQR(0:MaxItr) ! Overall fit Chi Square result
REAL CHIT(MaxTrk) ! Track contribution to Chi Square
REAL CHIV(0:MaxVtx) ! Vertex fit Chi Squares
REAL PAR (5,MaxTrk) ! Fitted track helix parameters
REAL PARDIF(5,MaxTrk) ! Parameter differences (fit - input)
REAL FMCDIF(MaxMcn) ! Mass constraint residuals
REAL PCON(MaxVtx,2) ! Pointing constraint tests
REAL SANG(MaxVtx,2) ! Pointing constraint tests
REAL XYZVRT(3,0:MaxVtx) ! Primary, Secondary vertices
REAL TrkP4(MaxTrk,6) ! Track Px,Py,Pz,E,Pt P
REAL VtxP4(4,MaxVtx) ! vertex 4-momentum sum
REAL McnP4(4,MaxMcn) ! 4-momentum sum for mass constraint
REAL DDA(MaxTrk,8) ! d(Px,Py)/d(Crv,Phi,Xs,Ys)
REAL DXYZPV(3) ! Primary vertex displacement in fit
INTEGER VOFF(MaxVtx),TOFF(MaxTrk) ! offset pointers into VMAT
INTEGER POFF(MaxVtx),COFF(MaxVtx) ! offset pointers into VMAT
INTEGER MOFF ! offset pointers into VMAT

"Memory"
INTEGER TKERR (MaxTrk) ! Track error flag, from previous fit
INTEGER IJKERR(3) ! error code reporting

DOUBLE PRECISION VMAT(MAXDIM,MAXDIM+1) ! Covariance matrix, fit parameters

REAL XYZPV(3)
EQUIVALENCE (XYZPV,XYZVRT(1,0))

INTEGER UVWXYZ(UDIM)
Equivalence (UVWXYZ, Iter)


CTC first approximation and track-vertex step parameters
REAL DRMAX, RVMAX, DZMAX, TRNMAX, DSMIN
Momentum scale conversion factor, from (cm)**-1 to Gev/c
REAL PSCALE
```

Figure A.2: *ctvmfq and ctvmfr common blocks (output).*

```
single precision stuff
COMMON /CTVMFq/ RUNNUM,TRGNUM, ITER,NTSCUT
,NVERTX,NMASSC,NTRACK ! Fit specification
,TRKVTX,TRKMCN,VTXPNT,CMASS,CVTX ! ...Fit specification
,VTXVTX ! vertex geneology
,TKBANK,LIST,TMASS ! Tracks to use
,MATDIM ! Fit matrix dimension
,TKERR ! Track "error memory"
,NDOF, CHISQR,CHIT,CHIV ! Fit dimension, quality
,XYZPV0,EXYZPV ! initial "primary" vertex
,XYZVRT,DXYZPV ! vertex coordinate results
,TRK0 ! Original track values
,COV0 ! Original track covariances
,PAR,G ! Track fit results
,TrkP4,VtxP4,McnP4,DDA ,VOFF,TOFF,POFF,COFF,MOFF ! vertex,track VMAT pointers
,PAR0,PARDIF ! Input parameters, fit step
,FMCDIF,PCON,SANG ! Mass/Pointing satisfaction
,DRMAX,RVMAX,DZMAX,TRNMAX,DSMIN ! CTVMFA constants
,IJKERR ! error code results
,PSCALE ! set, CTVMFT first entrance

double precision stuff
COMMON/CTVMFr/ VMAT ! Fit parameter covariance
```

Figure A.3: *ctvmfq and ctvmfr common blocks (blocks summary).*

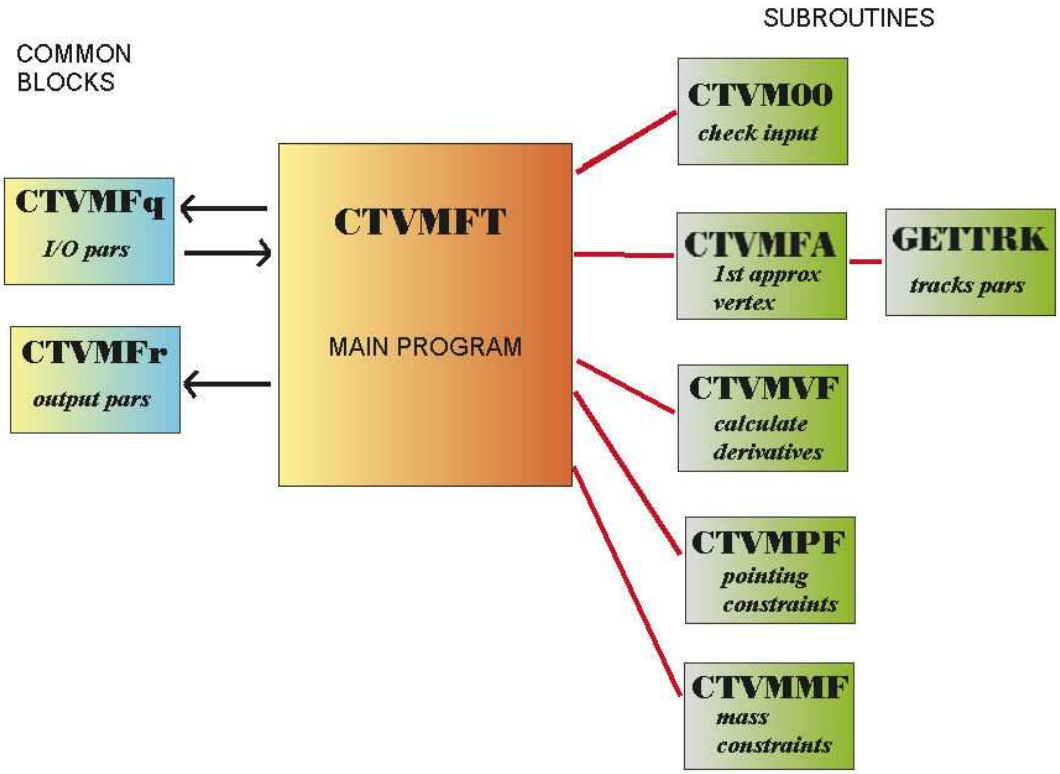


Figure A.4: *Scheme of CTVMFT routines and common blocks.*

```
int m_Nvtx;          number of fitted vertices
int m_Ntr;           number of tracks
int m_Npresel;       number of preselected tracks
int m_NTrue;         number of true tracks
int m_NprTrue;       number of preselected true tracks
float m_x;           vertex x position
float m_y;           vertex y position
float m_z;           vertex z position
float m_C2;          vertex χ^2
int m_Ndof;          number of degree of freedom
float m_C2rid;       reduced χ^2
float m_SPx;         summed momentum at the vertex (x comp.)
float m_SPy;         summed momentum at the vertex (y comp.)
float m_SPz;         summed momentum at the vertex (z comp.)
float m_SE;          summed energy at the vertex
double m_SigmaX;     error upon x position
double m_SigmaY;     error upon y position
double m_SigmaZ;     error upon z position
double m_CovXY;      covariance xy matrix element
double m_CovXZ;      covariance xz matrix element
double m_CovYZ;      covariance yz matrix element
int m_Kind;          kind of vertex:primary or secondary
float m_PrimX;       simulated primary vertex x position
float m_PrimY;       simulated primary vertex y position
float m_PrimZ;       simulated primary vertex z position
float m_BX;          simulated secondary vertex x position
float m_BY;          simulated secondary vertex y position
float m_BZ;          simulated secondary vertex z position
int m_secflag;       secondary vertex number
std::string m_p_rec; name of reconstruction package
list<SimpleTrackCollection::const_iterator> *m_VTlist;      fitted tracks collection
list<SimpleTrackCollection::const_iterator> *m_TruePlist;   collection of fitted tracks pointing to the primary vertex
list<SimpleTrackCollection::const_iterator> *m_TrueSlist;   collection of fitted tracks pointing to the secondary vertex
SimpleTrackCollection *m_ModT;      modified tracks collection
```

Figure A.5: *Members of SimpleVertex class.*

```
int M_FIT_MODE;          mode to use: only primary vertex, only secondary or both
double M_C2_CONTR;       maximum χ^2 contribution allowed (only for primary)
double M_C2_CONTRSEC;    maximum chi^2 contribution allowed (only for secondary)
bool M_NO_LOOP1;         flag to inhibit χ^2 loop (only for primary)
bool M_NO_LOOP2;         flag to inhibit χ^2 loop (only for secondary)
int M_SIM_INFO;          flag to activate the access to simulated info
int M_FLAGG;             flag to store the fit result (=1 if fit OK)
int M_M3;                flag to allow iteration both on primary and secondary vertices
int M_SECFLAG;           flag to allow multiple secondary vertices in the same event
int M_FLAGLOOP;          flag to break the loop over the tracks to check χ^2
int M_MTRK;              maximum number of tracks allowed
int M_MVTX;              maximum number of vertices allowed
int M_MMCN;              maximum number of constraints allowed

parameters to control the fit
int M_NmassC[MaxV];      number of mass constraints
float M_ExyzPv[3][3][MaxV];      covariance matrix
int M_VtxPnt[2][MaxV];           pointing vertices
float M_XyzPv0[3][MaxV];         hardwired primary vertex position
int M_Cvtx[MaxV];        conversion fit specifications
float M_Cmass[MaxC][MaxV];       mass constraints
```

Figure A.6: *Members of Flags_n_Par class.*

```
Flags_n_Par FnP; // Object of flags and parameters
Pars pars;
SetFnP(FnP,pars, Nevent, Nrun);

Vf_temp Vt; // Object to handle temporary vertex information

VfSIM Vs; // Object to handle simulated info

int n_Seeds = 0;
VfSeed SEED[m_num_secV];
FnP.set_M3(0); // used in MODE 3 to do first primary and then secondary v.
FnP.set_SECFLAG(0); // flag to iterate over secondary v.

ObjectVector<SimpleVertex> *Vcoll = new ObjectVector<SimpleVertex>; //vector of vertices
```

Figure A.7: *VertexExampleScheme: initialization of objects and flags.*

```
if(m_Fit_Mode != 1 && m_ext_track_list == 0){
Vs.GetSeed(pars, n_Seeds, SEED);
}

if(m_do_vertex_fit){

const DataHandle< SimpleTrackCollection > vec;
if (StatusCode::SUCCESS != m_store_gate->retrieve(vec) ) {
log << MSG::ERROR << "Could not find ContainedObject" << endreq;
return( StatusCode::FAILURE);
}

//

.....

//

do{ // do: used when looking for both primary and secondary vertex

list<SimpleTrackCollection::const_iterator> *MyTracks = new list<SimpleTrackCollection::const_iterator>;
list<SimpleTrackCollection::const_iterator> *SelectedTracks = new list<SimpleTrackCollection::const_iterator>;

ClearVt(Vt);

TrackSelection(FnP, MyTracks, SEED, vec, SelectedTracks);

if(m_get_sim_info == 1){
Vs.TrackSim(SelectedTracks, Vt, FnP, SEED );
}

DoFit(FnP,Vt,SelectedTracks);

FillVertex(FnP, Vt, Vcoll); // Fill vertex object

SelectedTracks->erase(SelectedTracks->begin(), SelectedTracks->end());
delete SelectedTracks;
MyTracks->erase(MyTracks->begin(),MyTracks->end());
delete MyTracks;

}while(FnP.SECFLAG() <= n_Seeds);

StatusCode sc = m_store_gate->record(Vcoll, m_vertex_location);
if( sc.isFailure() ) {
log<<MSG::ERROR<<"Error while registering vertex"
<<" with key " << "/Event/Vertex" << endreq;
return sc;}
} else {log << MSG::INFO << "Skipping Vertex Fit" << endreq;}
```

Figure A.8: *VertexExampleScheme: call of methods.*

```
// — Load shared libraries
ApplicationMgr.DLLs += { "VertexPrimary" };

// — Top algorithm

ApplicationMgr.TopAlg += { "VertexPrimary" };

//
// —– Vertex algorithm
//

VertexPrimary.OutputLevel = 1;

// Properties for vertex fit
// GENERAL properties
VertexPrimary.DoVertexFit = true; // true: do the fit
VertexPrimary.PatReco = "XKAL"; // Pettern Recognition type
VertexPrimary.FitMode = 1; // 1 only primary vtx; 2 only
secondary vtx; 3 both; VertexPrimary.VertexLocation = "Vertex";
//VertexPrimary.RetrieveVertex = 0; // Enable the retrieving and print out
of the vertex after storing in the TDS

// Primary Vertex properties
VertexPrimary.A0ImpactParameter = 0.03; // in cm
VertexPrimary.PTCut = .5; // pT cut
VertexPrimary.NumSiHits = 9; // Number of precision hits
VertexPrimary.BlayerHits = 1; // B-layer hit: 1 = yes 0 = no
VertexPrimary.NumPxlHits = 2; // Number of pixel hits
VertexPrimary.NoLoop1 = false; // true inibit Chi2
selection for primary vtx
VertexPrimary.C2Contr = 0.1; // C2/ndof of each track contribute to vertex
```

Figure A.9: *VertexPrimary jobOptions.txt file.*

```
// — Load shared libraries

ApplicationMgr.DLLs += { "VertexExampleSecSeed" };

// — Top algorithm

ApplicationMgr.TopAlg += { "VertexExampleSecSeed" };

//
// —– Vertex algorithm
//

VertexExampleSecSeed.OutputLevel = 1;

// Properties for vertex fit
// GENERAL properties
VertexExampleSecSeed.DoVertexFit = true; // true: do the fit
VertexExampleSecSeed.PatReco = "XKAL"; // Pettern Recognition type
VertexExampleSecSeed.FitMode = 3; // 1 only primary vtx; 2 only secondary vtx; 3 both;
VertexExampleSecSeed.VertexLocation = "Vertex";
VertexExampleSecSeed.RetrieveVertex = 0; // Enable the retrieving and print out
of the vertex after storing in the TDS

// Primary Vertex properties
VertexExampleSecSeed.A0ImpactParameter = 0.03; // in cm
VertexExampleSecSeed.PTCut = .5; // pT cut
VertexExampleSecSeed.NumSiHits = 9; // Number of precision hits
VertexExampleSecSeed.BlayerHits = 1; // B layer hit: 1 = yes 0 = no
VertexExampleSecSeed.NumPxlHits = 2; // Number of pixel hits
VertexExampleSecSeed.NoLoop1 = true; // true inibit Chi2 selection for primary vtx
VertexExampleSecSeed.C2Contr = 0.1; // C2/ndof of each track contribute to vertex

// Sec. Vertices properties
VertexExampleSecSeed.NumSecVtx = 2; // Maximum Number of sec vertex to be looked for if mode=3
VertexExampleSecSeed.NoLoop2 = true; // true inibit Chi2 selection for secondary vtx
VertexExampleSecSeed.C2ContrSec = 0.5; // C2/ndof of each track contribute to SECONDARY vertex
VertexExampleSecSeed.GetSimInfo = 1; // 1 -> get sim. info about vertices
VertexExampleSecSeed.A0ImpactParSec1 = 0.001; // for sec. v.: ParSec1 < ParSec2
VertexExampleSecSeed.A0ImpactParSec2 = 0.1; // for sec. v.
VertexExampleSecSeed.PTCutSec = .5; // pT cut
VertexExampleSecSeed.DeltaR = 1.4; // DR around seed
VertexExampleSecSeed.NumConstrSec = 0; // # of mass constraints used to fit secondary v.
VertexExampleSecSeed.ConstrMass1 = 1.; // Constraint #1 mass
VertexExampleSecSeed.ConstrMass2 = 1.2; // Constraint #2 mass
VertexExampleSecSeed.ConstrMass3 = 2.; // Constraint #3 mass
VertexExampleSecSeed.ConstrMass4 = 1.8; // Constraint #4 mass
VertexExampleSecSeed.ExternTrackList = 0; // Enable the use of a user define track list
to be used in the sec. v. fit; 0 == use of seeds;
```

Figure A.10: *VertexExampleSecSeed jobOptions.txt file.*

```
// — Load shared libraries

ApplicationMgr.DLLs += { "VertexExampleSecUserTracks" };

// — Top algorithm

ApplicationMgr.TopAlg += { "VertexExampleSecUserTracks" };

//
// —– Vertex algorithm
//

VertexExampleSecUserTracks.OutputLevel = 1;

// Properties for vertex fit
// GENERAL properties
VertexExampleSecUserTracks.DoVertexFit = true; // true: do the fit
VertexExampleSecUserTracks.PatReco = "XKAL"; // Pattern Recognition type
VertexExampleSecUserTracks.FitMode = 2; // 1 only primary vtx; 2 only secondary vtx; 3 both;
VertexExampleSecUserTracks.VertexLocation = "Vertex";
// VertexExampleSecUserTracks.RetrieveVertex = 0; // Enable the retrieving and print out of the vertex after
storing in the TDS

// Sec. Vertices properties
VertexExampleSecUserTracks.NumSecVtx = 4; // Maximum Number of sec vertex to be looked for if mode=3
VertexExampleSecUserTracks.NoLoop2 = true; // true inhibit Chi2 selection for secondary vtx
// VertexExampleSecUserTracks.C2ContrSec = 0.5; // C2/ndof of each track contribute to SECONDARY vertex
VertexExampleSecUserTracks.GetSimInfo = 0; // 1 -> get sim. info about vertices
// VertexExampleSecUserTracks.NumConstrSec = 1; // # of mass constraints used to fit secondary v.
VertexExampleSecUserTracks.ExternTrackList = 1; // Enable the use of a user define track list to be used in the
sec. v. fit; 0 == use of seeds
```

Figure A.11: *VertexExampleSecUserTracks jobOptions.txt file.*

```
package VertexFit

author Fabrizio Ferro <ferro@ge.infn.it>

use AtlasPolicy AtlasPolicy-*
use CxxFeatures CxxFeatures-00-* Utilities
use AgeToCxx AgeToCxx-* Tools
use GaudiInterface GaudiInterface-01-* External
use StoreGate StoreGate-02-* Control
use SimpleTrack SimpleTrack-00-* Reconstruction

use genz genz-00-*
use ggenz ggenz-00-*
use geant3 geant3-21-*

use CTVMFT CTVMFT-00-00-* External
use SimpleVertex SimpleVertex-00-00-* Reconstruction/Vertexing

library VertexFit *.cxx

apply_pattern installed_library
apply_pattern component_library



private

macro_remove use_linkopts $(VertexFit_linkopts)
```

Figure A.12: *VertexFit requirement file.*

```
package SimpleVertex

author Fabrizio Ferro <ferro@ge.infn.it>

use AtlasPolicy AtlasPolicy-*
use CxxFeatures CxxFeatures-00-* Utilities
use GaudiInterface GaudiInterface-01-* External
use StoreGate StoreGate-02-* Control
use SimpleTrack SimpleTrack-00-* Reconstruction

library SimpleVertex *.cxx

apply_pattern installed_library
apply_pattern component_library

private

macro_remove use_linkopts $(SimpleVertex_linkopts)
```

Figure A.13: *SimpleVertex requirement file.*

```
package VertexPrimary

author Fabrizio Ferro <ferro@ge.infn.it>

use AtlasPolicy AtlasPolicy-*
use CxxFeatures CxxFeatures-00-* Utilities
use GaudiInterface GaudiInterface-01-* External
use StoreGate StoreGate-02-* Control
use ZebraTDREvent ZebraTDREvent-00-* Event
use EventAthena EventAthena-00-* Event
use SimpleTrack SimpleTrack-00-* Reconstruction
#use AthenaCommon AthenaCommon-* Control

use SimpleVertex SimpleVertex-00-00-* Reconstruction/Vertexing
use VertexFit VertexFit-00-00-* Reconstruction/Vertexing

library VertexPrimary *.cxx

#apply_pattern installed_library
apply_pattern component_library
```

Figure A.14: *VertexPrimary requirement file.*

```
package VertexExampleSecSeed

author Fabrizio Ferro <ferro@ge.infn.it>

use AtlasPolicy AtlasPolicy-*
use AgeToCxx AgeToCxx-* Tools
use CxxFeatures CxxFeatures-00-* Utilities
use GaudiInterface GaudiInterface-01-* External
use StoreGate StoreGate-02-* Control
use ZebraTDREvent ZebraTDREvent-00-* Event
use EventAthena EventAthena-00-* Event
use EventStructure EventStructure-00-* Event
use SimpleTrack SimpleTrack-00-* Reconstruction

use genz genz-00-*
use ggenz ggenz-00-*
use geant3 geant3-21-*


use SimpleVertex SimpleVertex-00-00-* Reconstruction/Vertexing
use VertexFit VertexFit-00-00-* Reconstruction/Vertexing

library VertexExampleSecSeed *.cxx

#apply_pattern installed_library
apply_pattern component_library
```

Figure A.15: *VertexExampleSecSeed requirement file.*

```
package VertexExampleSecUserTracks

author Fabrizio Ferro <ferro@ge.infn.it>

use AtlasPolicy AtlasPolicy-*
use AgeToCxx AgeToCxx-* Tools
use CxxFeatures CxxFeatures-00-* Utilities
use GaudiInterface GaudiInterface-01-* External
use StoreGate StoreGate-02-* Control
use ZebraTDREvent ZebraTDREvent-00-* Event
use EventAthena EventAthena-00-* Event
use SimpleTrack SimpleTrack-00-* Reconstruction

use genz genz-00-*
use ggenz ggenz-00-*
use geant3 geant3-21-*

use SimpleVertex SimpleVertex-00-00-* Reconstruction/Vertexing
use VertexFit VertexFit-00-00-* Reconstruction/Vertexing

library VertexExampleSecUserTracks *.cxx

#apply_pattern installed_library
apply_pattern component_library
```

Figure A.16: *VertexExampleSecUserTracks requirement file.*

Bibliography

[1] http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/architecture/General/index.html

[2] http://tarta.home.cern.ch/tarta/vtx/docu.html