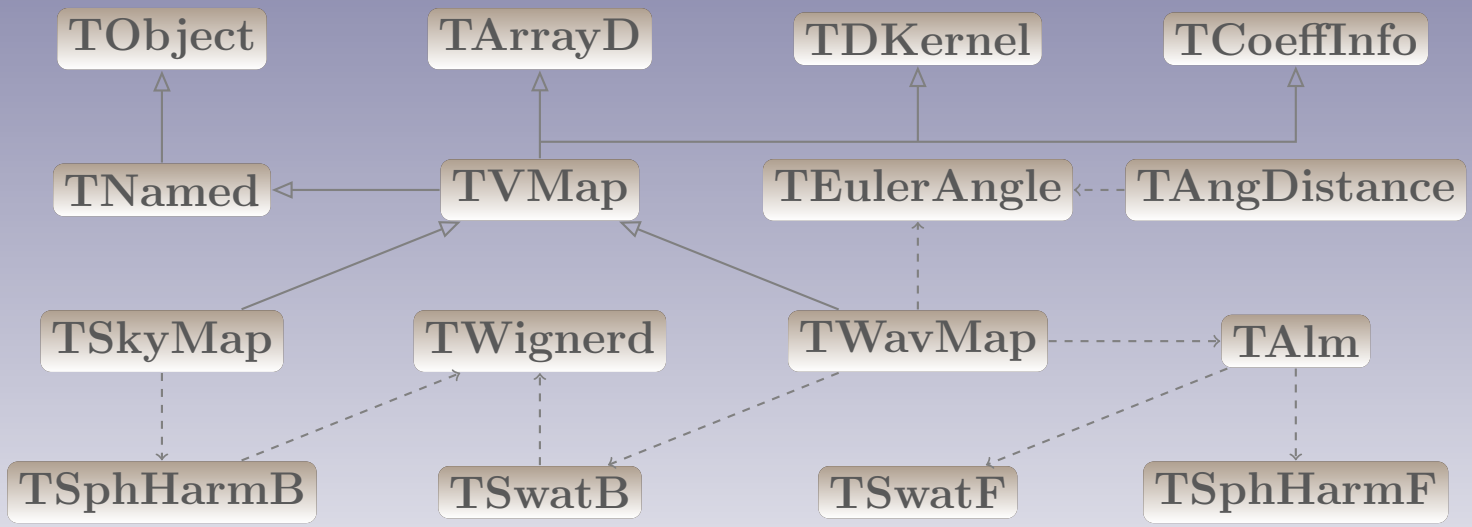
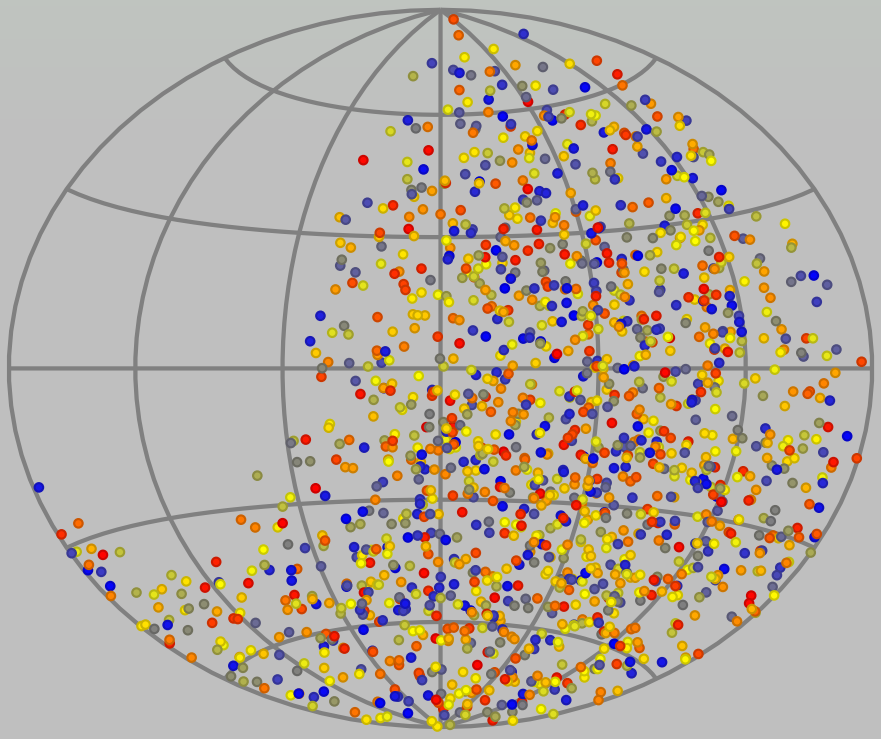



# SWAT

## The Spherical Wavelet Analysis Tool



Manual for version 1.44 - Marcelo Zimbres

# Contents


**1 Introduction** **2**
- 1.1 Installation . . . 3
- 1.2 Getting ready . . . 4
- 1.3 High Quality graphs with pgfplots . . . 4

**2 The three main programs** **5**
- 2.1 swat . . . 5
- 2.2 swatfind . . . 5
- 2.3 swatsim . . . 6

**3 Deflection vs. Energy** **6**

**4 Counting in stripes** **7**

**5 Wigner-d functions** **7**

**A swatsim** **7**


# 1 Introduction

SWAT is a package for analysis of functions that live on the sphere, which I have mostly used to analyze data of the Pierre Auger experiment and CRPropa simulations. I have written it from scratch as part of my my Ph.D. project. It evolved from a simple ROOT macro and got bigger and bigger, since the code may be useful for other people I decided to organize, document it and make it public. The code is an C++ implementation of the spherical wavelet transform as presented in [2]. Some attractive features of SWAT are:

- Harmonic and Wavelet transform on the sphere.
- Interface to Healpix code, which is included in the build system.
- Easy selection of events hitting sky window.
- Calculation of deflection vs. $1/E$ graphs.
- Calculation of the Wigner-d functions via FFT.

- ROOT and FFTW are the only prerequisite.

The package includes some parts of the Healpix code. There are two reasons why I decided to include it here, instead of just link against Healpix libraries.

1. I do not need all Healpix routines and support to the fits format.
2. I usually need shared libraries to call Healpix code in a ROOT session, which is not built by Healpix build system since most of its code are C++ templates.

Most of the theoretical datails of analysis of functions defined on $S^2$ and $S^3$ were taken from [2, ?, ?].
The code has been tested in LINUX and MACBOOK, but the code is portable enough to be built on other platforms. In the following we describe how to use SWAT. Questions concerning the software can sent to Marcelo Zimbres mzimbres@gmail.com

## Acknowledgements

Most of SWAT were developed with financial support from CAPES. Additionaly I would like to acknowledge the Institute of Physics at UNICAMP and the Bergische Universität Wuppertal for finacial support, Brunel University for granting me a scholarship to participate in the Cern School of Computing, the Ettore Majorana Foundation and Center for scientific Culture, the brasilian group of the Pierre Auger experiment, with special thanks to my supervisor Ernesto Kemp, for beliving in the project and Rafael Alves Batista for testing the code. The Pierre Auger Group at university of Wuppertal, with special thanks to Nils Niertenhöfer for helping me with CRPropa.

## 1.1 Installation

As a prerequisite to install SWAT you need ROOT and FFTW installed, the configure script will complain if they are not installed. SWAT uses autotools to generate configure and the Makefile, so you can expect all the standard configure options and makefile targets. The lines bellow will install the libraries on "/usr/local"

```
$ ./configure
$ make
$ make install
```

To use some of the features of the package, I am assuming you are able to build shared libraries on your platform. To easy the task of loading swat libraries the macro load.C, macro will be installed on "prefix/share/swat". To load the libraries in ROOT's C++ interpreter you just have to execute it in your ROOT session, or just add it to your code .rootlogon.C macro.

```
$ cp /usr/local/shar/swat/load.C ~/.rootlogon.C
$ root # The .so's are automatically loaded here.
```

All SWAT classes are now available in the ROOT session with syntax highlighting. You should also be able to generate documentation in HTML format with the macro "prefix/share/swat/makehtml.C". You have to run this macro from the directory where you built SWAT. The documentation will be built in the directoryhtmldoc.

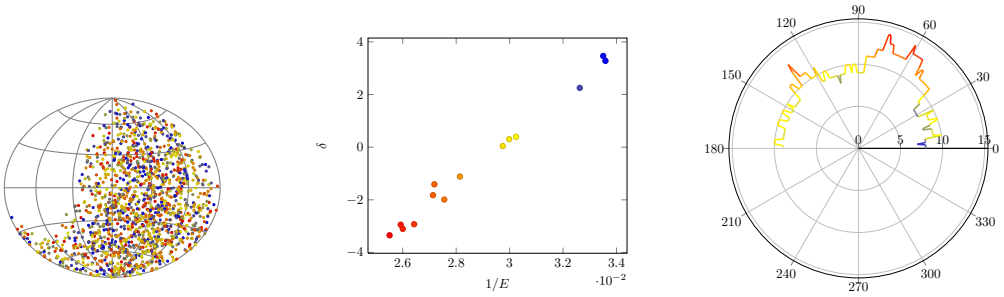
## 1.2 Getting ready

To analise Herald data, you will have to convert the ascii file to a TTree and save it in a .root file, for that you should use the macro prefix/share/swat/-convert_herald.C where prefix is usually */usr/local*. Copy this macro to the same directory where you have the herald data file. The macro will read a file with name "herald.dat", so you may have to rename it.
For CRPropa simulations you can just configure CRPropa to output a .root file instaed of a text file. The code uses the title of the TTree to diferentiate between a Herald and CRPropa file. The title of the TTree for CRPropa must be *"CRPropa 3D events"*.

## 1.3 High Quality graphs with pgfplots

If you have pgfplots installed on your machine, you can use the LaTeXfiles which are available in the directory pgfplots to generate high quality graphs. You may have noted that after running swatfind you get many text files in your working directory. They are input files for the LaTeXfiles. Here are some

example graphs




# 2 The three main programs

The main programs SWAT will install are:

## 2.1 swat

swat: This program only used to benchmark and test the algorithm. It can test both the spherical harmonic transform and the spherical wavelet transform.

## 2.2 swatfind

swatfind: This is the main program. It can be used to find sources and for cosmic ray data also multiplets. It saves source locations and energy-deflection graphs in root format, that can be inspected later. Main output is printed on the screen. A source here is a term used to mean something with a position on the sky, usualy described by $(\theta, \phi)$ and an orientation. We will use the three Euler angle $(\alpha, \beta, \gamma)$ to describe the source. You can use the following code to find sources in your data. For example:
The algorithm goes as follows:

1. The TTree is read selecting events with energies in the range $emin < E < emax$.

2. A Healpix map is generated and transformed to wavelet space. The parameter $N$ gives the precision on the ability of the wavelet to find the angle $\gamma$ and $j$ is the scale at witch the wavelet transform is performed. Refer to [**?**] for the meaning of these parameters.

3. A partial sort is used to find the nsources first biggest wavelet coefficients. The position of the wavelet coefficient, described by the three euler angles is saved in a TEulerAngle object and saved in the file "sources.root"

Once you have a file with the sources objectss, you can see the source position using the method TEulerAngle::Show

```
$ root sources.root
root[1] gDirectory->ls()
TFile**         sources.root
TFile*          sources.root
KEY: THealpixMap        hmap;1  Healpix sky map
KEY: TEulerAngle        source0;1
KEY: TEulerAngle        source1;1
KEY: TEulerAngle        source2;1
root [2] source0->Show(1)
wav(137.812,-27.4219,159.638) = -0.035312
root [3]
```

### 2.3 swatsim

swatsim: Uses Monte Carlo simulations to calculate the probability of a multiplet happen by chance on a isotropic sky. We still do not have means of simulating exposure.

## 3 Deflection vs. Energy

You can also easily calculate graphs of deflection versus $1/E$ this way:

```
root[1] TAuxFunc::find_multiplets(1,w,"sources.root","chain.root")
```

This code will:

1. Read the sources file "sources.root"
2. For each source it will calculate the equations descibing a stripe of length $l$, width $w$ tangent to the sources and aligned with it.
3. Read the Herald data, select all events hitting the stripe and calculate the deflection versus $1/E$ graphs.
4. The graphs will be added to the sources file.

```
$ root sources.root
root[1] gDirectory->ls()
TFile**         sources.root
TFile*          sources.root
KEY: THealpixMap    hmap;1  Healpix sky map
KEY: TEulerAngle    source0;1
KEY: TEulerAngle    source1;1
KEY: TEulerAngle    source2;1
KEY: TEulerAngle    source3;1
KEY: TGraphErrors   g0;1    C = 0.216, N = 22
KEY: TGraphErrors   g1;1    C = 0.294, N = 15
KEY: TGraphErrors   g2;1    C = -0.013, N = 23
KEY: TGraphErrors   g3;1    C = -0.005, N = 42
root [2] g0->GetCorrelation()
root [2] g0->Fit("pol1")
root [2] g0->Draw("ap")
```

The graph with name $g0$ correspond to source $source0$ and so on. The function of the last section will also calculate the "Number of events vs. stripe orientation" and add to the sources.root file. The graphs will be named $g0$, $g1$ ...

# 4 Counting in stripes

# 5 Wigner-d functions

# A swatsim

```
Calculates the probability of a multiplet with minimum correlation C (see -c
option) and minimum number of events m (see -m option) happen by chance,
using wavelet analysis. First an isotropic sky is simulated and the wavelet
representation of the sky is calculated, the euler angles of the largest
coefficient is used to calculate the equations of the tangent plane at the
position found (the euler angles). The correlation C is calculated including
all events that hit the tangent plane, whose size is specified with the
options -l and -w. The probablility will be the number of multiplets with
correlation > C and number of events > m, diveded by the number of skies
simulated. Additionaly, two other quantities are calculated, the histogram
of the number of events that hit the tangent plane and the histogram of the
C's found for which the number of events is greater than m (passed in the
command line). If -f option is used, TTree in the file will be read and
events will be added to the analysis, this is useful to include a simulated
multiplet on the analysis, hiding it in the isotropic backgroung the test the
algorithm.

Usage: swatsim [ -j scale] [-N number] [-n nevents] [-s skies] [-i emin]
       [-e emax] [-c corr] [-m mevents] [-w width] [-l length] [-f file.root]

   Options:

-h:     This menu.
-j:     Wavelet scale a number in the range 0 <= j <= 8, defaults to 1.
-N:     Band limit of wavelet, in the range 0 < N <= 128, defaults to 1.
-n:     Number of events in the simulated sky, defaults to n = 1000
-s:     Number of skies to simulate, defaults to 100.
-i:     Minimum energy of events, defaults to 20 EeV.
-e:     Maximum energy of events, defaults to 40 EeV.
-c:     Minimum correlation, defaults to 0.2.
-m:     Minimum number of events hitting tangent plane.
-w:     Width of tangent plane, defaults to 2 degrees.
-l:     Length of tangent plane, defaults to 10 degrees.
-f:     Add events in TTree stored in file to the simulated sky.
```

# References


[1] J Fourier Anal Appl (2008) 14: 145179.

[2] Mon. Not. R. Astron. Soc. 000, 122 (2007).

[3] http://www.ifi.unicamp.br/~mzimbres/