Statistical Analysis Methods for Physics Models Verification and Validation

Silvia De Luca[1]*, Marilena Bandieramonte[2], Witold Pokorski[2]

Abstract
The validation and verification process is a fundamental step for any software like Geant4 and GeantV, which aim to perform data simulation using physics models and Monte Carlo techniques. As experimental physicists, we have to face the problem to compare the results obtained using simulations with what the experiments actually observed. One way to solve the problem is to perform a consistency test. Within the Geant group, we developed a C++ compact library which will be added to the automated validation process on the Geant Validation Portal.

Keywords
Validation — GeantV — Geant4 — Consistency Test — GoF

[1] *Department of Physics, University of Bologna, Bologna, Italy*
[2] *EP-SFT, CERN, Geneva, Switzerland*
***Corresponding author**: silvia.deluca8@studio.unibo.it

Contents

Introduction

Geant4 [1], first released in 1998, is a software package designed tto simulate the interaction between particles going through matter. It contains all the aspects of the simulation process from the geometry of the system to the visualization of the particle trajectories. To test and validate each Geant4 release, the collaboration developed a web tool: Geant Validation Portal [2]. Now, the validation and verification routine takes place; during the software development process, along with the improvement of the models to describe the underlying physics of the simulations, it is crucial to have some evidence of the modifications. For example, we want to answer questions like:

- How well does Geant4 describe the data from experiments?
- Which physics model, provided by Geant4, describes data best?

The answer comes from a detailed study of the overlaid plots comparing for example two versions of the tool, the same version but with different physics list and so on. In 2012, the GeantV project started [3][4]. It tries to reach the same goal of Geant4 with different means. The GeantV simulation prototype introduced the concept of vectorized processing in particle transport. An α version of GeantV will be released during the end of the year. The main improvements were:

- Vectors;
- Level Parallelism and micro-parallelism;
- Hardware threading;
- Multi-core and multi-node;

where a lot of effort is spent on further development. Hence, due to the rapid growth, there is the need to have a method to perform these comparisons quickly; within the GeantV group was already developed a validation routine [5]. The project described in these pages, focuses on performing consistency tests using the statistic quantity p-value as result of a Goodness-of-fit test. The data used to test the code are for the moment just simulated, used for unit tests on algorithms. In the next pages, we are going to explain from both the theoretical and experimental point of view the algorithm operational meaning. We will explain also the code structure and how to use it.

1. Methods

1.1 P-value and Goodness-of-fit

The P-value was first defined within the statistics field of hypothesis testing. What we have now is a sample of observed data, some parameters and we know nothing about their actual distribution. Usually one formulates and initial hypothesis H_0, the *null hypothesis*, to be compared with others alternative hypothesis. Each hypothesis can be either simple or composite, our case is the simple hypothesis since every parameters is known; along with the hypothesis is specified a pdf $f(x|H)$ which could describe the data. We can perform with the H_0 hypothesis either a hypothesis test, say, H_0 against just one alternative hypothesis H_1 or a GoF (Goodness-of-Fit) test, H_0 against any other alternative hypothesis. Within this work we are interested in the GoF test. In this context to accept or reject the H_0 hypothesis we have to formulate a *test statistic* t; it is a function of the data and it is constructed in a way that should point out if the data are compatible with H_0 or not. The quantity P-value is a way to state about the consistency of data with respect to a certain hypothesis. It is defined as:

$$p = \int_t^{\infty} f(t^{'}|H)\,dt^{'} \tag{1}$$

By definition "*The P-value is a quantity defined for a hypothesis H and it is the probability under assumption of H, to observe data with equal or lesser compatibility with that hypothesis relative to the data we got.*" [6]
As one can see from (1), the P-value is a function of the data hence a variable; it means that under the *null hypothesis* it will follow a certain distribution. In our study we focused more on this important property of the P-value: when the null hypothesis is true, any p-value between 0 and 1 is equally likely. When the alternative hypothesis is true, p-values near 0 are more likely. A p-value near 0 therefore gives evidence that the alternative hypothesis holds.

$$g(p|H) = 1 \Rightarrow 0 \leq p \leq 1 \tag{2}$$

But, to give meaning to the P-value in the decision to accept or reject the hypothesis being tested, we have to put a threshold on its value: the *significance level* α. It is the probability to accept the *null hypothesis* when it is false; then we want this probability to be very small.

$$P(p \leq \alpha|H) = \alpha \text{ and if } P-value \leq \alpha \tag{3}$$

we can reject the hypothesis. Frequently used α values are: $0.01, 0.05$ or 0.1.

1.2 Test statistics

In this subsection we want to describe the test statistics being used; as stated before, the test statistics are useful because they can indicate whether the data are consistent with the hypothesis or not. This property is due to their mathematical definition that treats each test statistic features differently. Four test statistics have been selected: χ^2, Kolmogorov-Smirnov, Anderson-Darling and Cramér-Von Mises.

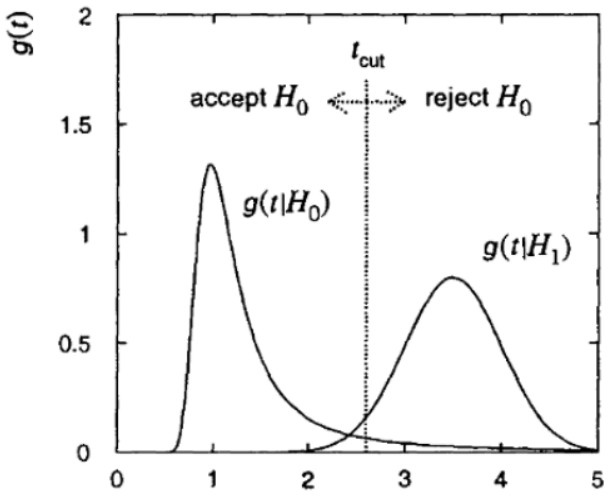

Figure 1. This picture shows the distribution of a test statistic g given two hypothesis H_0 and H_1 and the threshold on a t value of the test statistic to reject or accept H_0.

the tests used are the so called *two-samples test.*

Pearson's χ^2 In literature exist several definitions of χ^2, for our scope we choose the one given by Pearson [7]. This test is formulated only for binned samples, but it is not actually a limit, because for unbinned data one can discretize the sample generating an histogram, but this procedure implies a information loss. One has to keep into account that the values of the chi-square test statistic are dependent on how the data is binned. Another disadvantage of the chi-square test is that it requires a sufficient sample size in order for the chi-square approximation to be valid (each bin must have more than 5 entries). χ^2 formula is:

$$\chi^2 = \sum_{i=1}^{N}\sum_{j=1}^{2}\frac{(n_{ij} - E_{ij})^2}{E_{ij}} \tag{4}$$

where N is the number of observations, n_{ij} is the observed value i for the sample j, E_{ij} is the expected value of n_{ij} and the denominator is the square of variance ($sqrtE_{ij}$).

Kolmogorov-Smirnov The test is based on the maximum difference between an empirical and hypothetical cumulative distribution [8]: $D_K = sup_x(|F_n(x) - F(x)|)$.
Within the code the version used is:

$$T_{KS} = sup_x(|F_n(x) - G_m(x)|) \quad (5)$$

where we are comparing two empirical cumulative distribution for two samples (x_n) and (x_m). This test is used to verify the null hypothesis that the two samples are taken from the same events flow. One relevant property of the test is that we do not need to know which is the common distribution. Then, the KS is distribution-free only if the H_0 hypothesis deals with independent observations which are distributed with a continuous distribution. In principle the KS applies only to continuous distribution. This is important because each data are treated separately and there is no information loss; another characteristic is that it is more sensitive to deviations near the center of the distribution than at the tails.

Anderson-Darling Using this test statistic we are considering the difference on the quadratic distance. The Anderson-Darling distance [9] is a specialization of KS. It was first developed for detecting sample distributions' departing from normality, its formula is:

$$T_{AD} = n\int_{-\infty}^{\infty} [F(x) - F_n(x)]^2 \psi(F(x))\, dF(x) \quad (6)$$

$$\text{with } \psi(F(x)) = [F(x)(1 - F(x))]^{-1}$$

Now the difference is that the weight function ψ confers to the test more sensitivity to the deviations at the tails. It can be performed both on binned and unbinned data, as before, also this test is distribution-free and non-parametric. A notable feature is that AD is better capable of detecting very small differences even for large samples.

Cramér-Von Mises likewise the Anderson-Darling test statistic, the CVM [10] concerns the difference on the quadratic distance and derives from the same formula (6) but in this case the weight function ψ is equal to 1:

$$T_{CVM} = n\int_{-\infty}^{\infty} [F(x) - F_n(x)]^2\, dF(x) \quad (7)$$

The test is based on the assumption that the empirical distribution functions being compared are unbinned, but the test is conservative also for binned. Less powerful than the Anderson-Darling test, the CVM applies on both symmetric and right-skewed distributions. CVM is distribution-free and non-parametric test for H_0 hypothesis of independent observations.

2. Code structure

Overview

As already expressed in the introduction, this project comes as an attempt to fulfill the need for a quick comparison between data; we have to consider the P-value as a trigger for our analysis. Several scenarios may occur and the algorithms to compute the P-value may behave differently, then we need to pay attention to which we want to use. In this section we will touch two main topics: the preliminary work done and the code implementation.

2.1 Preliminary work

Our target was already fixed, we wanted to perform consistency tests on *data*. We choose to use the test statistics introduced in the section above; as data we mean both *binned* and *unbinned*. We wanted to manage histograms, graphs, graphs with error on the y values (binned) and continuous distributions (unbinned). We started our study from two already existent tools: ROOT [11] and Statistical Toolkit [12-14]; both of them provide a wide set of tools for statistical analysis, in particular the Statistical Toolkit contributes with plenty of statistical algorithms for data comparison. For our experimental need, we wanted to create a small and compact pool of tests; moreover, without lots of dependencies, these tests have to adapt to the different types of inputs data we expect.

2.2 Implementation

The histograms, graphs and graphs with error on the y values are represented in ROOT, respectively as: TH1F/D, TGraph and TGraphErrors; while the unbinned data type is represented using the std vector. These data format are actually used to store informations in the validation database, for example:

Histograms we have TH1F/D, F and D stand for float or double precision and the choice is arbitrary, as output of regression tests (say verification routines);

Graphs TGraph is used for experimental data, sometimes we can also find the format **TGraphErrors** or TH1F/D;

Unbinned are seldom used, but basically represent some random variable following a continuous distribution.

	DATA			
METHODS	TH1	TGraph	TGraphErrors	std::vector
χ^2	ROOT	X	new	X
A-D	ROOT	new	X	ROOT/new
K-S	ROOT	new	X	ROOT/new
C-VM	S-T	new	X	new

Tab1. Summary of the implemented methods and the relative accepted data type.

In the table above we are summarizing all the algorithms used and to which type of input data can apply and if it is a newer or previous implementation. To make clear the table above, let's consider the input type *TH1D*: when we find the *X* it means that there is not possibility to use the test on this type of data,otherwise it is specified if we use an already existent version (ROOT or Statistical Toolkit) or a new one. For what

concern the Statistical Toolkit, it is a library based on an old analysis tool AIDA;for example he Cramér-Von Mises (CVM) implementation follows the one in the Statistical-Toolkit (ST). For compatibility sake we only used ROOT. As shown, the χ^2 cannot be used with TGraph or std::vector since we need to associate an error to the data. The label *ROOT/new* means that already exist an implementation in ROOT for unbinned data, but it deals only with pointers; the new version can accept std::vectors directly, but the result is the same.

The library we developed contains four classes, each of them is implemented for just one test statistic; each class derives from one parent class, which is an abstract class used only to provide a common interface for the functions dealing with the input data. Its derived classes implement their own version of the method and for the computation of *distance* there are as many overloads as the input they can receive, some of them just call the already existing ROOT method. When the algorithm is called, we extract the informations from the input data, like: their entries, number of bins, x-values, y-values, errors on y-values etc. All these informations are stored inside different std::vectors, then we use this final form to compute the test statistic value.

Four different Unit Tests have been implemented. They aim at stressing the algorithms and highlight their behaviors under modifications of the initial distribution which may consist of rigid translations or deviations restricted to a small area. They represent a preliminary study on the power of the tests.

2.2.1 Anderson-Darling

The class AndersonDarling implements two functions to compute the value of the distance, one for TGraph and the other for std::vector; in the case of TH1 (both double and float precision) it calls the ROOT implementation. Therefore, the distance function [15] has three overloads. The non-trivial ones:

TGraph in this case the distance function takes as arguments two TGraph, the first operation is to extrapolate two std::vector, for both the TGraph, filled respectively with values on the x-axis and y-axis. Within the computation we may consider the TGraph likewise an histogram: the x-values correspond to the bin center and the y-values as the multiplicity of that value. Then we have to estimate the cumulative distribution, which is defined as: $F_j = \sum_{i=-\infty}^{x_j} \frac{x_i}{N}$. Where x_i stands for the multiplicity of a certain $x < x_j$, and N is the total number of entries, say the sum of all the y-values.
The cumulative for each spot in the graph is stored inside a std::vector. The important thing in this chain is that the two graphs must have the same values on the x-axis; in fact, let's say that we are comparing two graphs obtained measuring the cross section for different values of the energy; to do so, we must have the same energy values for the histogram, but it doesn't mean that we should observe always the same values on y-axis. Moreover it is important to underline that the x-values can also be not equally spaced. Then, the distance value obtained is sent to the function for P-value computation.

std::vector for unbinned distribution the main difference consists of how we compute the cumulative distribution and what we consider now the relevant data. The function in this case receives two std::vector, the samples (x_i) and (y_j), which are first sorted and then sent to the function for the cumulative. We consider that inside the sample there are no repetitions, we tested the algorithm with random data (about a million) coming from, for example, a Gaussian distribution, so it is quite unlikely to have the same value for different trials.
So, the cumulative is obtained assigning to each entry a weight of 1; in this way N is effectively the number of points. For the distance and P-value is used the same algorithm used for TGraph.

For the P-value we used the following formula [9]:

$$P(T^2{}_{AD} \leq z) = \frac{\sqrt{2\pi}}{z} \sum_{j=0}^{\infty} \binom{-\frac{1}{2}}{j} (4j+1) e^{-\frac{(4j+1)^2\pi^2}{8z}} \int_0^{\infty} e^{\frac{z}{8(w^2+1)} - \frac{(4j+1)^2\pi^2 w^2}{8z}} dw \tag{8}$$

it is a one-sample limit distribution, where z is the observed value of the test statistic.

2.2.2 Pearson's χ^2

The χ^2 test has been implemented only for TGraphErrors, for what concerns histograms is already existent in ROOT; this class has the same methods used for TGraph to get values on the x-axis, on the y-axis but also a method to get the y-errors (nErrors) and a function to compute the sum of squared errors (sum_of_2errors) weighted for the y-value which in our approximation of a graph with an histogram corresponds to the bin content. The routine to compute the χ^2 distance has been modified from the one existent in ROOT; we used the approach for the comparison between two weighted histograms [16] that in our case can be modified to accept TGraphErrors, in the simple case the error that one can use is the squared root of the bin content; then, the approximation of the test is:

$$\chi^2 = \sum_{i=1}^{N} \frac{(W_1 w_{2i} - W_2 w_{1i})^2}{W_1{}^2 s_{2i}{}^2 + W_2{}^2 s_{1i}{}^2} \tag{9}$$

where W_1 and W_2 correspond to the total sum of the weight of the first graph and the second one, while w_{ji} is the weight value relative to ith entry in the graph , s_{ji} is the variance

estimator of w_{ji}- with weights we mean the y-values which we treat as the multiplicity related to a bin with center equal to the corresponding x-value. The P-value is given using the incomplete complementary gamma function [17]:

$$Q(x,a) = \int_x^\infty e^{-t}t^{a-1}\,dtx \geq 0 \tag{10}$$

with x equal to the observed χ^2 and a as the degree of freedom.

2.2.3 Kolmogorov-Smirnov

For this test statistic we implemented the Kolomogorov-Smirnov class that has an overloaded function (to accept three different arguments) to calculate the distance. For histograms and unbinned distributions the function just calls the correspondent methods in ROOT. It has been added an implementation for the TGraph:

TGraph in this case we compute the cumulative directly inside the distance function since it consists of few elements; as before each component of the cumulative std::vector is normalized to the total amount of entries (we consider the multiplicity). The main operation is the search for the maximum value of the difference between the two cumulative distributions. When the function is called, an intermediate difference is compared with the previous one already calculated, until the maximum value is found. Then, to calculate the P-value associated, we use the ROOT function *KolmogorovProb* which takes as argument the normalized value of the distance (normalized to the size of the two samples); it calculates the probability to exceed the actual value of the distance using the Kolmogorov's distribution assuming the null hypothesis true:

$$P(z) = 2\sum_{j=1}^{\infty}(-1)^{j-1}e^{-2j^2z^2} \tag{11}$$

where z is the observed value of the test statistic [16].

2.2.4 Cramér-Von Mises

For this statistic test [18], both the algorithms for histograms, graphs and unbinned data are implemented. The operational way is the same utilized in the case of Anderson Darling. We have different functions that provide us informations relative to the number of bins, the value of the bin center and etc. For all the three computations the cumulative is calculated in the same way for histogram and tgraph, since the assumption already explained for the Anderson-Darling, and for the unbinned data the cumulative is computed assigning a weight 1 to each entry;

TH1 as before, the final form in which the data are managed is a std::vector, in this case (likewise for graph) we do not need to sort the data along the x-axis, and neither we need to consider centers of the bins equally spaced. we only have the constraint to give the same x-values for both histograms.

The P-value we get is computed using a limiting distribution that is the same for both two-samples and one-sample test, using the modified Bessel function of the second kind ($K_{1/4}$) for an observed value z of the test statistic [19]:

$$P(T^2{}_{CVM} \leq z) = \frac{1}{\pi\sqrt{z}}\sum_{j=0}^{\infty}(-1)^j \binom{-\frac{1}{2}}{j}\sqrt{(4j+1)}e^{-\frac{(4j+1)^2}{16z}}$$
$$K_{1/4}[\frac{(4j+1)^2}{16z}] \tag{12}$$

2.2.5 Unit Test

The Unit Tests, developed and used in this code, focus on different aspects of the distribution and on the initial assumption we make. They are four and are listed below:

NB: to explain the tests we use as example the implementation for histograms which is more straightforward. Tests are implemented also for graphs and unbinned data, the idea is the same.

TEST 0 this Unit Test aims to prove the important property of the P-value being uniformly distributed between [0,1] if the null hypothesis is true. To do so, we simulated some data starting from an initial distribution (we used a Gaussian and Uniform); this test exists for the cases: histogram, graph, graph with error on y-values and unbinned. There are two ways to perform the test.
The first way is to start with two separated histograms filled with data extracted randomly and separately from the same distribution; the result is two histograms that have slight differences in the bin multiplicity. The other way is to start from the same histogram filled randomly, for example, with a Gaussian and then, from this, we fill others two histogram always randomly. We have to follow the rule of independent observations.
This test is performed using a MC generation of the histograms' content.

TEST 1 once we observed that the P-value follows the expected distribution, we are also interested in testing if it behaves correctly when the two distributions depart from being consistent with the null hypothesis.
Now the final result is graphically showing the trend of the observed P-value values. Now the final result is graphically showing the trend of the observed P-value values. To get this output we start with the generation process of the two histograms to be compared, always

randomly filled (see Fig.2); then we compute several times the P-value, each time we compare a couple of random histograms; we repeat this process many times. At each repetition we increment the value of the mean of the distributions used to fill the histograms of a constant step and leaving the other parameters fixed.
Then, we store the actual value of the step and the mean of P-value distribution; the final plot shows on the x-axis the values of the step and on the y-axis the correspondent (mean) P-value. Accordingly to this test, we start from two overlapping distribution and at each iteration we are shifting the two distributions. Therefore we expect to start from a P-value of 0.5 and that gradually goes to 0.

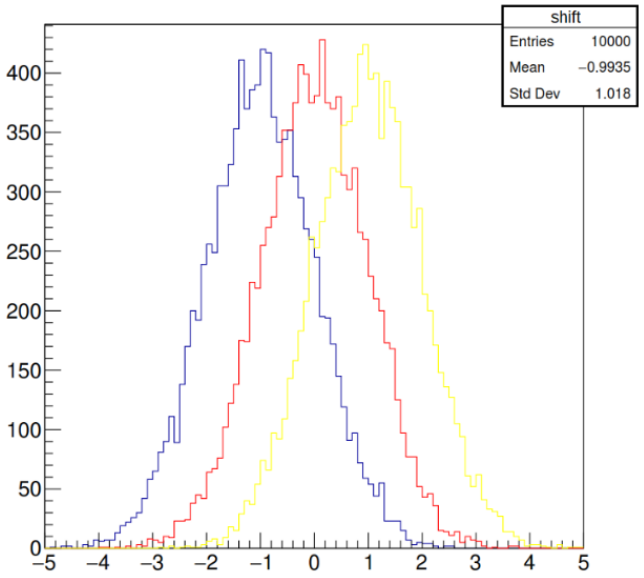

Figure 2. This picture shows the shift applied to the initial distribution (red line) on the left (blue) and on the right (yellow); at each step we compare the blue distribution with the yellow one.

TEST 2 this test aims at verify the trend of P-values when the process of modification is different. In this case we compare a fixed histogram (A) with one randomly filled (B). The filling process is subdivided into three steps: first we get the 40% of the entries from two Gaussian distributions G_1 and G_2 (20% from G_1 and 20% from G_2), the remaining 60% of entries comes from the initial histogram.
Now, we modify the mean of G_1 and G_2 and leave fixed the sigma; the total amount of entries is always the same.
Then, we shift the mean values by a constant step to the left and to the right. So, we compare the first histogram (the one fixed) with the modified one; The final modified distribution shows a significant difference in the tails then in the sigma (see Fig.3). In the final plot we always show the mean P-value and the shift value; Therefore we expect to start from a P-value of 0.5 and that gradually goes to 0.

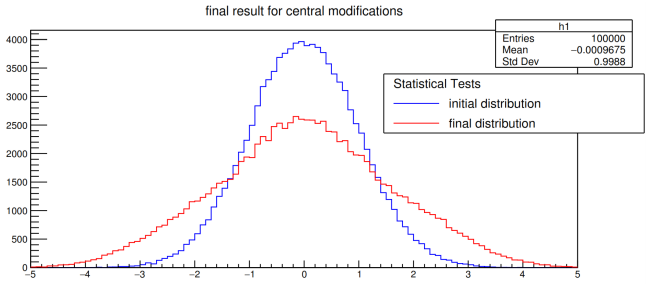

Figure 3. Here it is shown the final result obtained with TEST 2; at the last iteration we compare the blue line (initial fixed distribution) with the red one (modified).

TEST 3 this test is similar to TEST 2 but the value modified is the sigma of G_1 and G_2 added.
We compare a fixed histogram (A) with one randomly filled (B). The filling process is subdivided into three steps: first we get the 40% of the entries from two Gaussian distributions G_1 and G_2 (20% from G_1 and 20% from G_2), the remaining 60% of entries comes from the initial histogram.
Now, we modify the sigma of G_1 and G_2 and leave fixed the mean; the total amount of entries is always the same.
Then, we shift the sigma values by a constant step to the left and to the right. So, we compare the first histogram (the one fixed) with the modified one; The final modified distribution shows a less significant difference in the tails but more evident at the center (see Fig.4). In the final plot we always show the mean P-value and the shift value; Therefore we expect to start from a P-value of 0.5 and that gradually goes to 0.

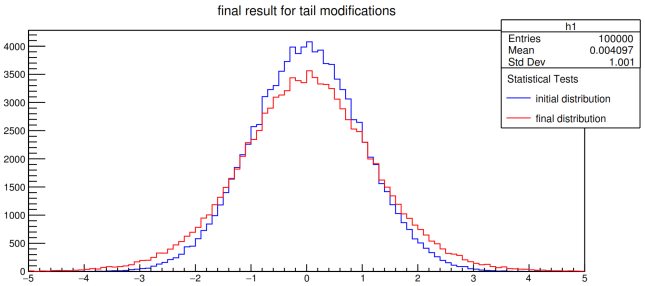

Figure 4. Here it is shown the final result obtained with TEST 3; at the last iteration we compare the blue line (initial fixed distribution) with the red one (modified).

3. Results and Future Work

In this section we present some results obtained with TGraph as an example.

TGraph

TEST 0

As explained before, the Unit Test 0 aims to prove the uniform distribution of the P-Value. The results obtained for TGraph are shown in figures 5, 6 and 7; These histograms have been produced through a MC simulation of 10^5 pseudo-experiments, each one with 1 million events. In Fig.5 one can

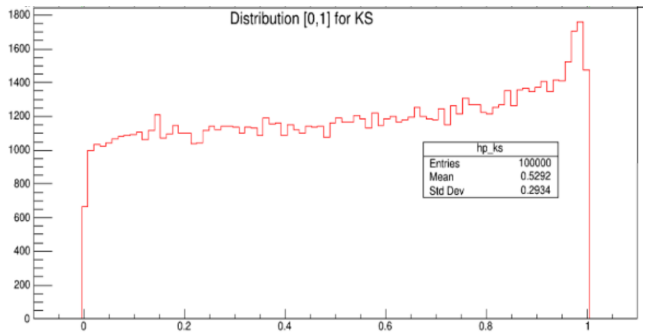

Figure 5. P-value distribution for Kolmogorov-Smirnov test statistic.

see the distribution obtained using the Kolmogorov-Smirnov test statistic for P-value; as already said, this test statistic is thought for unbinned data. Then, when we use it on binned data we have to pay attention to the number of bins. It must be big in respect with the number of entries; in this case we have 10000 bins. So, the result we got, in this situation, is biassed by theory; the distribution is expected to tend to be uniform when the bin width is really small ($<<1$). In Fig.6

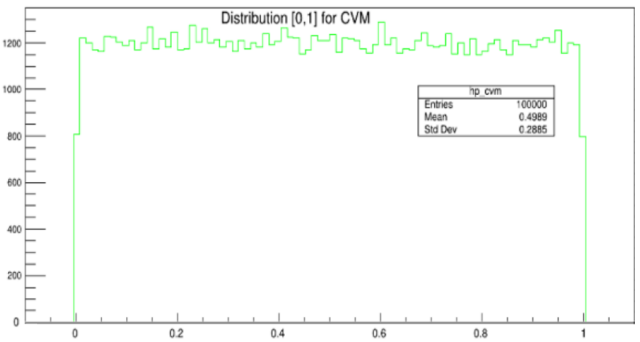

Figure 6. P-value distribution for Cramér-Von Mises test statistic.

we can see results for the CVM test. The amount of pseudo-experiment is the same as before as well as the number of entries and bins. An ideal scenario should be a distribution completely flat, this result could be obtained increasing the number of pseudo-experiments. Fig.7 shows the result of the test when using the AD test. It presents a ”trend” in P-value

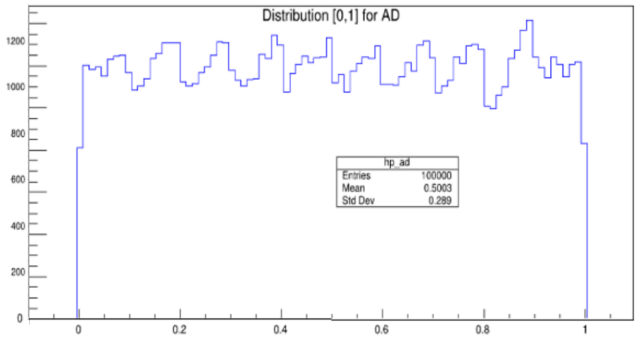

Figure 7. P-value distribution for Anderson-Darling test statistic following the ROOT implementation.

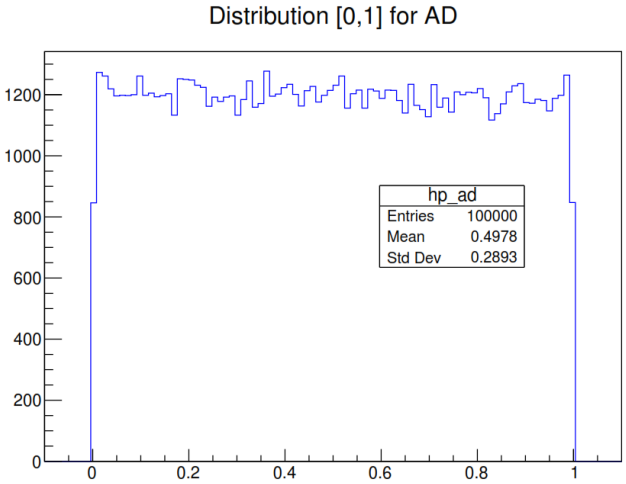

Figure 7.1.P-value distribution for Anderson-Darling test statistic following the Statistical Toolkit implementation.

distribution because in this case we used the ROOT implementation that uses tabulated values to calculate the P-value; in Fig.7.1 it is shown the same distribution obtained this time using a limiting function to compute the P-value, as one can see there is no pattern.

TEST 1

Using this test we want to show which test is faster to recognize the inconsistency between the two distributions being shifted; each point represents the mean P-value (over 10000 values). We are shifting the value of the mean of the distribution, at the first step the shift is equal to zero, so the two graphs (always generated randomly) are coming from the same events flow and we got 0.5 P-value; As stated, the KS gives values higher than AD or CVM, this is because of its theoretical definition. The green line (AD) and the red line (CVM) are the fastest to go to zero, but to a more careful look, one can see that already for a shift in the mean of 0.002 the AD is more sensitive than CVM. The trend is very similar, even if biased as stated in theory.

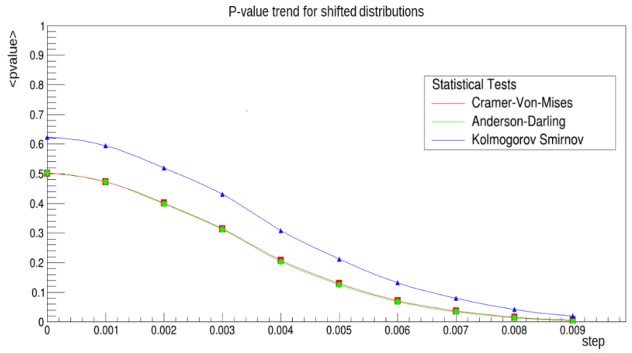

Figure 8. Results of TEST 1: P-value trend. Cramér-Von Mises (red), Anderson-Darling (green) and Kolmogorov-Smirnov (blue).

TEST 2

As already shown in Fig.3 the comparison performed in TEST 2 consists of an evident discrepancy between sigmas of the two distributions; in figure Fig.9 one can see effectively that already for a step of 0.06 the Anderson-Darling statistic is more sensitive since the modifications are more centered at the tail region; for the first iteration we obtain a value higher than 0.5, probably due to a fluctuation in the production of the two graphs within the pseudo-experiment.

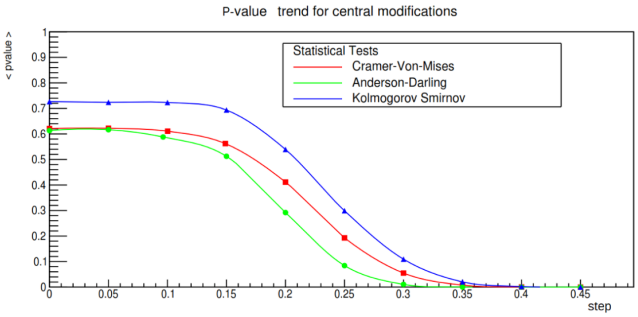

Figure 9. P-value for TEST 2 using Cramér-Von Mises, Anderson-Darling and Kolmogorov-Smirnov. Each point represents the mean value of a sample of 10^4 values.

TEST 3

In this case we add two distributions with a variable sigma and then compare the initial sample with the modified one. The AD is still more sensitive than the KS and CVM, as before the KS gives higher values than AD and CVM; the AD by definition is more sensitive to differences within the tail region, we do not see really well here this aspect because adding other two distribution with variable sigma to a fixed one does not change the location of major statistic, the data are always centered within the mean value; same for the CVM, more sensitive to symmetric distribution, the symmetry of the distribution is not changed by this operation; only the KS is more sensitive to center deviation, but still there is the bias due to the binned data.

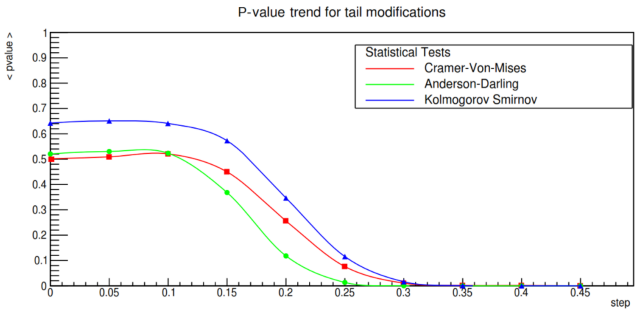

Figure 10. P-value for TEST 3 using Cramér-Von Mises, Anderson-Darling and Kolmogorov-Smirnov. Each point represents the mean value of a sample of 10^4 values.

3.1 Results: χ^2

For completeness we add the results obtained using χ^2 for TH1 and TGraphErrors.

The first result is related to TEST 0 showing the P-value uniformly distributed;

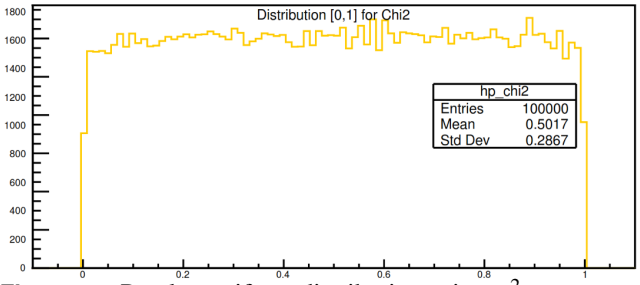

Figure 11. P-value uniform distribution using χ^2 on histograms

In Fig.11 it is shown the P-value distribution obtained using the ROOT implementation of χ^2 for histograms; as one can note in this case there is no pattern inside the values visible, in fact here the P-value is computed through a probability function and not tabulated values. In Fig.12 it is shown the result of χ^2 new implementation for TGraphErrors, which is visibly near to a uniform distribution. The following figures

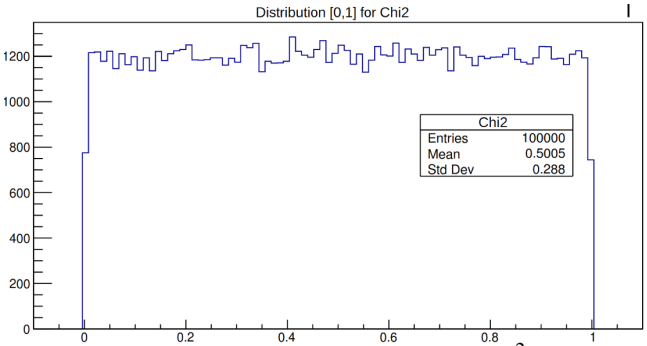

Figure 12. P-value uniform distribution using χ^2 on TGraphErrors.

13 and 14 show the trend of P-value for the shifting of distri-

butions; respectively, Fig.13 shows the results relative to TH1 and Fig.14 for TGraphErrors. In both cases we can see that the χ^2 slower to recognize the modification in distributions

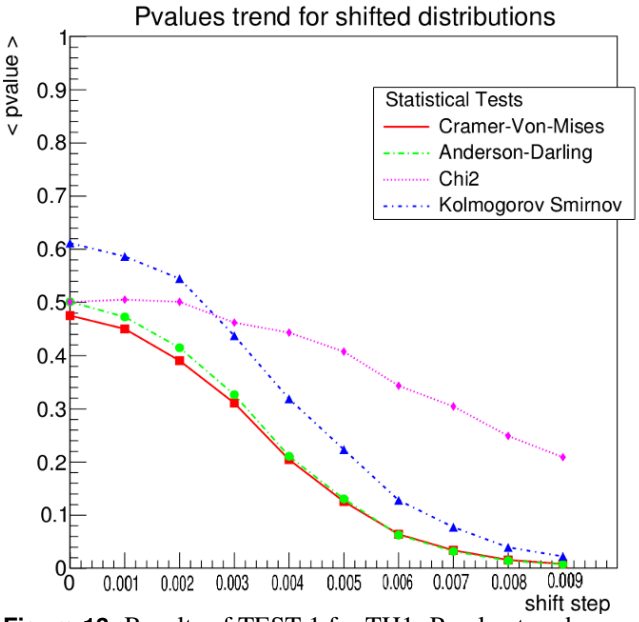

Figure 13. Results of TEST 1 for TH1: P-value trend. Cramér-Von Mises (red), Anderson-Darling (green), Kolmogorov-Smirnov (blue) and χ^2 (pink).

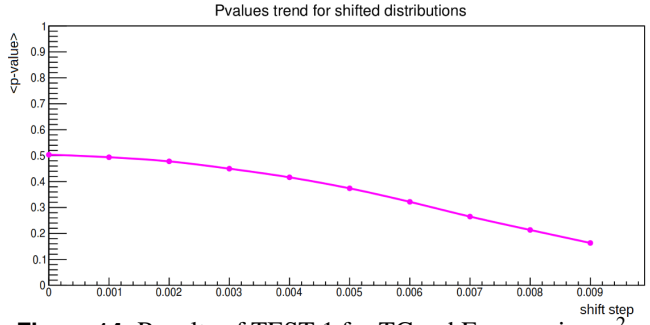

Figure 14. Results of TEST 1 for TGraphErrors using χ^2.

Future Work

There is still lot of work and more detailed studies to do on this really interesting topic.

Future effort will be put on the improvement of mean and and standard deviation of the P-value distribution; we tried to include different possible input case and test how the statistical tests behave. As theoretically stated CVM and AD are more sensitive to changes in distributions; also the KS is giving good results with some constraints on the number of bins.

It is not showed in this report but, the χ^2 test appears to be not sensitive ad CVM or AD and slower to reach 0.

Future work may also focus on adding a new test developed by Serguei Bityukov about the "significance of difference" [20].

References

[1] Geant4 official site, *http://geant4.cern.ch/applications /index.shtml*.

[2] Geant Validation Portal, *https://geant-val.cern.ch/*.

[3] GeantV website, *http://geant.cern.ch/content/about-geantv*.

[4] G Amadio et al. *"GeantV: from CPU to accelerators"*, Journal of Physics: Conference Series 762 (2016) 012019, *http://iopscience.iop.org/article/10.1088/1742-6596/762/1/012019/pdf*.

[5] G Amadio et al. "Verification of Electromagnetic Physics Models for Parallel Computing Architectures in the GeantV Project".

[6] G Cowan, 1998, *"Statistical data analysis"*.

[7] Agresti, A. (2007), *An Introduction to Categorical Data Analysis*, John Wiley & Sons Hoboken, NJ.

[8] M. A. Stephens, *"Use of the Kolmogorov-Smirnov, Cramer-Von Mises and Related Statistics Without Extensive Tables"*, Journal of the Royal Statistical Society. Series B (Methodological) Vol. 32, No. 1 (1970), pp. 115-122.

[9] F. W. Scholz and M. A. Stephens, *"k-Sample Anderson-Darling Tests"*,J.Amer. Stat. Assoc. 82 (1987) 918.

[10] T. W. Anderson, *"On the Distribution of the Two-Sample Cramér-Von Mises Criterion"*, Annals Math. Stat., 33 (1962) 1148.

[11] ROOT website,*https://root.cern.ch/*.

[12] Statistical Toolkit website, *https://www.ge.infn.it/statisticaltoolkit/*.

[13] Mascialino et al, *"New developments of the goodness-of-fit Statistical Toolkit", IEE Transactions on Nuclear Science, Vol. 53, no.6, 2006;* Mascialino et al, *"New developments of the goodness-of-fit Statistical Toolkit", IEE Transactions on Nuclear Science, Vol. 53, no.6, 2006;*

[14] G. A. P. Cirrone et al, *"A Goodness-of-Fit Statistical Toolkit*, INFN/AE-04/08, 2004.

[15] F Porter, *"Testing Consistency of Two Histograms"*, Pasadena, California 91125 March 7, 2008.

[16] N.D. Gagunashvili, *"Chi-Square Tests for Comparing Weighted Histograms"*, arXiv: 0905.4221v3, 2009.

[17] G.J.O. Jameson, *"The incomplete gamma functions"*, Cambridge University Press.

[18] Eadie et al, *"Statistical Methods in Experimental Physics"*, pp (269-270).

[19] M. Rosenblatt, *"Limit theorems associated with variants of the von Mises statistic", Anls. Ma. St. vol.23, pp.617-623, 1952*.

[20] S Bityukov, *https://indico.cern.ch/event/567550/contributions/2660835/attachments/1508337/2351994/Bityukov_2017_S.pdf*.