

## MEMORANDUM

**From/De** : Nicolas Pierre  
**To/à** : COMPASS collaboration  
**Subject/Sujet** : DJANGO : a Monte-Carlo Generator with Radiative Corrections

This note explains the functioning of the DJANGO Monte-Carlo generator (H. Spiesberger, generator of ep interaction at HERA modified for COMPASS needs), its structure and its specificities, as well as reporting results obtained by the generator and the COMPASS Monte-Carlo chain after being coupled to TGEANT.

Also available at <http://wwwcompass.cern.ch/compass/notes/2018-1/2018-1.html> .

## Contents

<b>1</b>	<b>Presentation of DJANGO</b>	<b>2</b>
<b>2</b>	<b>Introduction to radiative corrections</b>	<b>2</b>
<b>3</b>	<b>Radiative corrections in analysis</b>	<b>4</b>
3.1	Characterization and impact of radiative corrections . . . . .	4
3.2	About emission of radiative photons . . . . .	5
<b>4</b>	<b>Self-consistency of DJANGO</b>	<b>6</b>
<b>5</b>	<b>Comparison between DJANGO and RADGEN</b>	<b>8</b>
<b>6</b>	<b>TGEANT and DJANGO</b>	<b>9</b>
6.1	Forewords : Technical description of DJANGO . . . . .	9
6.2	Implementing DJANGO in TGEANT . . . . .	13
6.3	TDJANGO : a C++ interface to DJANGO . . . . .	15
<b>7</b>	<b>TDJANGO : How to ?</b>	<b>17</b>
<b>8</b>	<b>Radiative correction factors</b>	<b>18</b>
8.1	Inclusive radiative correction factors . . . . .	18
8.2	Semi-Inclusive radiative correction factors : Effect on Multiplicities . . . . .	20
<b>9</b>	<b>Conclusion</b>	<b>23</b>
<b>10</b>	<b>Acknowledgement</b>	<b>23</b>
	<b>References</b>	<b>24</b>

# 1 Presentation of DJANGO

## What is DJANGO ?

- DJANGO[1] is at first a Monte-Carlo event simulation tool for neutral and charged current  $ep$  interactions at HERA with the event generators HERACLES and DJANGO6.
- DJANGO was then modified to also simulate  $\mu p$  interactions at the COMPASS experiment.
- The emphasis is put on the inclusion of QED radiative corrections (single photon emission from the lepton or the quark line, self energy correction, complete set of one-loop weak corrections). The background from radiative elastic scattering  $\mu p \rightarrow \mu p \gamma$  is also included.
- HERACLES is treating the  $lp$  scattering by means of structure function parametrizations or parton distribution functions in the quark-parton model framework.
- DJANGO6 is simulating deep inelastic scattering including both QED and QCD radiative effects.
- DJANGO is an interface to LEPTO[2], ARIADNE[3] (for parton cascades), PYTHIA[4] (LUND string fragmentation in JETSET[5] for hadronic final state) and SOPHIA[6] (for low-mass hadronic final states).

## What can you do with DJANGO ?

- Generation of  $lp$  scattering with and without fragmentation for the final state with radiative events.
- Calculation of cross-sections (radiative, born)
- Calculation of radiative correction factors (inclusive, semi-inclusive)
- Use it as an event generator in a Monte-Carlo chain

# 2 Introduction to radiative corrections

Radiative corrections had a key role in the development of QED : they enable one to calculate cross-sections with extremely high precision that has until now not been contradicted by experiment. When a process like DIS involves charged particles, there is a rearrangement of the electromagnetic current between the initial and final state. Incidentally, real photons can be emitted.

If one were to compute the cross section of the process  $1 + 2 \rightarrow 3 + 4$  with no photon emission (Born level process), one would find a different result  $\sigma_0$  than the measurement  $\sigma_{exp}$ . To obtain a more accurate result, one has to consider the processes  $1 + 2 \rightarrow 3 + 4 + \gamma_1 + \gamma_2 + \dots + \gamma_n$ . Taking these corrections to Born level process into account, one obtains :

$$\sigma_{exp} = (1 + \delta_{RC})\sigma_0 \tag{1}$$

where  $\delta_{RC}$  are the radiative corrections.

In this note, I will not discuss the corrections beyond first order. These first order corrections are also known as order  $\alpha$  ( $o(\alpha)$ ) corrections. They comprise :

- Leptonic radiation
- Hadronic radiation (parton model)
- Interference of lepton/hadron radiation (two-photon exchange)
- Vacuum polarization
- Weak corrections

The goal is to quantify the effect of radiative corrections. The observed cross-section can be expressed as the convolution of the true cross-section times a function called the radiator function which takes into account the radiative effects :

$$d\sigma^{obs}(p, q) = \int \frac{d^3k}{2k^0} R(l, l', k) d\sigma^{true}(p, -q, k)$$

This relation holds also for the structure functions :

$$F_n^{obs}(x, Q^2) = \int d\tilde{x} d\tilde{Q}^2 R_n(x, Q^2, \tilde{x}, \tilde{Q}^2) F_n^{true}(\tilde{x}, \tilde{Q}^2)$$

The previous formulas are valid for one photon emission but can be extended to include higher-order multi-photon emissions. As one has access to both observed quantities and radiator function, the determination of the true cross-sections or structure functions from measured ones is done by unfolding using an iterative procedure. The principal drawbacks of such a method is that the solution is ill-defined : there is no unique solution, there are large uncertainties and the process is numerically unstable. However, using partial functioning on the radiator function is reducing the instability of the calculation :

$$R(l, l', k) = \frac{I}{k.l} + \frac{F}{k.l'} + \frac{C}{\tilde{Q}^2}$$

The partial functioning is splitting the radiator function in three :

- Initial state radiation ( $I$  fraction)
- Final state radiation ( $F$  fraction)
- Compton peak ( $C$  fraction)

For each of them, an observation can be made. For initial state radiation (ISR),  $k.l$  is small for  $\angle(l_{in}, \gamma) \rightarrow 0$ , for final state radiation (FSR),  $k.l'$  is small for  $\angle(l_{out}, \gamma) \rightarrow 0$  and eventually for Compton peak  $Q^2$  is small for  $p_T(l_{out}) \simeq p_T(\gamma)$ .

For ISR and FSR, the photon is emitted within narrow cones with a width of the order of  $\sqrt{\frac{m_t}{E_t}}$ . If the particle is massless, the radiated photon will be colinear to the lepton (incoming or outgoing, whether it is an ISR or a FSR).

Two last notes are to be made :

- As  $E_{\gamma, max}^2 \propto Q^2 \frac{1-x}{x}$ , the largest radiation in energy are at large  $Q^2$  and small  $x$ . Radiation is suppressed at small  $Q^2$  and large  $x$ . There are also large negative corrections from uncanceled virtual contributions.
- As  $\tilde{Q}_{min}^2 = \frac{x^2}{1-x} M_N^2$ , the case where  $\tilde{Q}_{min}^2 \ll Q^2$  is possible.

All the preceding explanations did concern leptonic radiation. One should also adress the question of the hadronic corrections (quark line radiation). These corrections are infrared divergent (radiation of soft photons and gluons) but they cancel with loops. The emission of the photon/gluon is collinear and gives rise to correction of type  $\frac{\alpha}{2\pi} \log(m_q^2)$ . Nevertheless for quarks, the approximation  $m_q \approx 0$  is giving rise to divergent corrections. One way to solve this issue is to factorize and absorb the divergences into the PDFs :

$$d\sigma = \sum_f d\hat{\sigma}_f(1 + \delta_f(Q^2, m_q^2)) q_f(x) = \sum_f d\hat{\sigma}_f \hat{q}_f(x, Q^2)$$

### 3 Radiative corrections in analysis

#### 3.1 Characterization and impact of radiative corrections

In the following, only DIS/SIDIS is considered.

The description of a radiative event is given by the following : an event is called radiative as soon as it contains one real radiated photon which is emitted in the lepton line (Fig. 1).

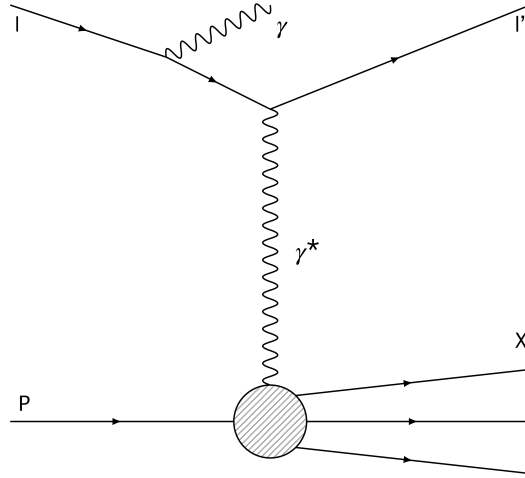


Figure 1: Typical feynman diagram of a radiative event. One can note that the pair  $(Q^2, \nu)$  at the vertex (called hadronic) is not the same as the one calculated using the incoming and outgoing lepton (called leptonic). The relation between the two pairs is drawn by :  $\nu_{had} = \nu_{lep} - E_\gamma$ ,  $Q_{had}^2 = Q_{lep}^2 + 2E_\gamma(\nu_{lep} - \sqrt{\nu_{lep}^2 + Q_{lep}^2} \cos\theta_\gamma)$

In the following, we will only consider these corrections (Fig. 2) :

- Internal Bremsstrahlung (from both incoming and outgoing leptons) (b,c)
- Vertex correction (d)
- Vacuum polarization (e)

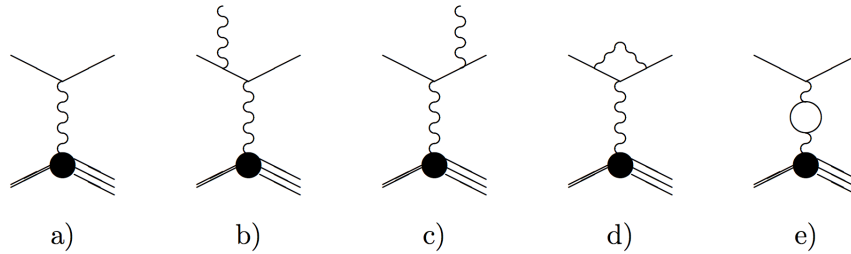


Figure 2: List of the diagrams used for the calculation of the radiative corrections. From left to right, tree level, internal bremsstrahlung (incoming and outgoing leptons), vertex correction and vacuum polarization.

Correction to the quark line are not included in any calculation, as explained in Section 8. If we call  $\sigma_{Born}$  the cross-section of the tree-level diagram and  $\sigma_{Born+o(\alpha)}$  the cross-section of tree-level plus the first order correction enumerated above, the definition of the radiative corrections factor  $\eta$  is :

$$\eta(x, y) = \frac{\sigma_{Born}(x, y)}{\sigma_{Born+o(\alpha)}(x, y)} \quad (2)$$

where  $x$  is the Bjorken- $x$  aka the fraction of momentum of the incoming proton carried by the struck parton and  $y$  is the fraction of energy of the incoming lepton transferred by the virtual photon.

Observation : the definition of  $\eta$  is not fixed and you can find other definition of it. You can especially encounter the following definition :

$$\eta(x, y) = \frac{\sigma_{Born+o(\alpha)}(x, y)}{\sigma_{Born}(x, y)} \quad (3)$$

which is the inverse of the one before. **Be extremely careful of the definition of  $\eta$  you are using as it can be very deceptive.** In this note, the former definition (2) will be used.

Obviously, the emission of a real photon is modifying the kinematic variables of the event : two events will share the same leptonic variables but they will have different hadronic variables. In the case of multiplicities, this discrepancy in the kinematic variables induces that some hadrons are falling into the wrong  $(x, y)$  bin. Applying the correction factor  $\eta$  to the multiplicities is indirectly redirecting the hadrons between  $(x, y)$  bins.

### 3.2 About emission of radiative photons

There are two privileged angles for emission of a real photon :

- One in the direction of the incident lepton (s-peak)
- One in the direction of the outgoing lepton (p-peak)

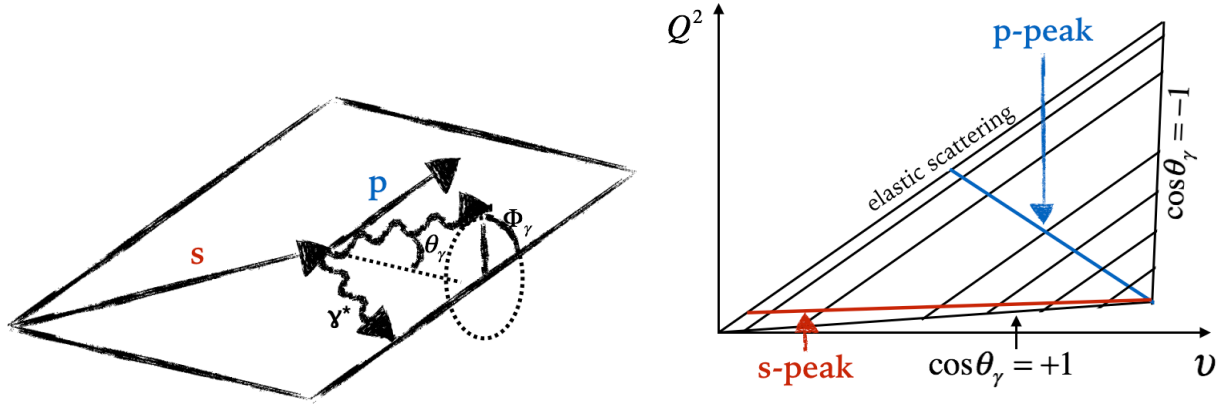


Figure 3: Angles characterizing the emission of a radiative photon :  $\theta_\gamma$  is the polar angle and  $\Phi_\gamma$ .

Figure 4: Location of the privileged angles in the  $(Q^2, \nu)$  plane. The phase space is delimited by the elastic scattering on top and the boundary values of  $\cos\theta_\gamma$ . The s-peak is located near the boundary  $\cos\theta_\gamma = 1$  which means  $\theta_\gamma \equiv 0[2\pi]$ , thus colinear to the incoming lepton.

In the case of muons, note that the s and p peaks are much less pronounced than for electrons.

These knowledge will later be useful to verify the consistency of DJANGO results.

## 4 Self-consistency of DJANGO

In order to test the self-consistency of DJANGO, I generated a certain number of events of  $\mu p$  scattering with an incoming muon energy of 160 GeV. In Fig. 5), the energy of the radiated photon (if one is present), the outgoing muon and the struck quark are shown. The cutoff at low energy is given by the specified kinematic cuts in DJANGO.

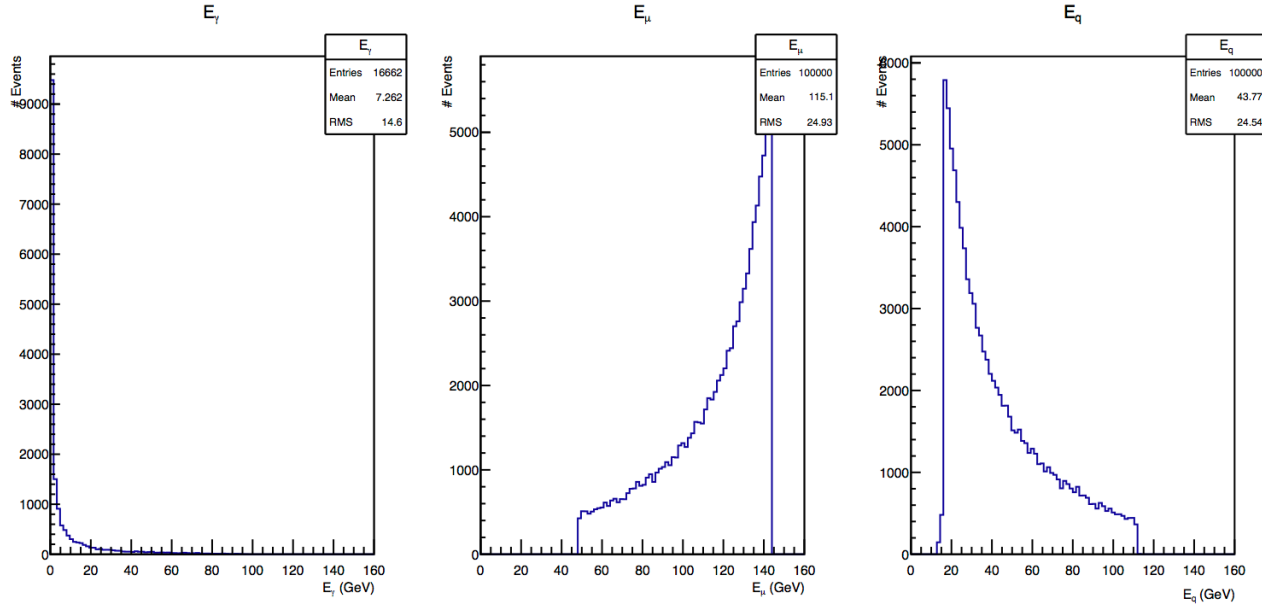


Figure 5: Energy distribution for, from left to right, radiative photon, outgoing lepton and struck quark. The emission of low energy radiative photon is privileged by DJANGO, while the rest of the energy is distributed between the outgoing muon and the struck quark.

We expect naively a peak around  $E_\gamma = 0$  as soft photons (low energy photons) are more likely to be emitted than hard photons (high energy photons). The rest of the energy of the incoming muon is distributing accordingly between outgoing muon and struck quark.

Fig. 6 shows the  $\theta_\gamma$  and  $\theta_\mu$  distributions. As discussed before, the radiated photon has two privileged direction of emission, namely the s-peak and the p-peak, colinear to the direction of propagation of the incoming and outgoing muons. In Fig. 6, the two distribution are plotted next to each other, enabling to see if the p-peak is matching with the position of the peak in the scattering angle of the outgoing muon, which is the case. The s-peak is around 0, which is also expected.

A last exercise that I have done is to see whether DJANGO, when we put a  $Q^2 > 1$  constraint on event generation, is producing radiative events with  $Q^2 < 1$ , which would be the case if a radiative photon was emitted by the incoming lepton. A quick proof of this can be made starting from the relation between  $Q_{lep}^2$  and  $Q_{had}^2$  :

$$Q_{had}^2 = Q_{lep}^2 + 2E_\gamma(\nu_{lep} - \sqrt{\nu_{lep}^2 + Q_{lep}^2} \cos\theta_\gamma)$$

When a radiative photon is emitted by the incoming muon,  $\cos\theta_\gamma \simeq 1$  then  $\nu_{lep} - \sqrt{\nu_{lep}^2 + Q_{lep}^2} \cos\theta_\gamma \leq 0$  leading to  $Q_{had}^2 \leq Q_{lep}^2$ . With an analogous reasoning, if a radiative photon is emitted by the outgoing photon, then  $Q_{had}^2 \geq Q_{lep}^2$ . In Fig. 7,  $Q_{had}^2$  is shown as a function of  $Q_{lep}^2$ , for  $Q_{lep}^2 = 1$ . Values of  $Q_{had}^2$  can be found both above and below  $Q^2 = 1$ , which was the point to be verified. Another information that is given by this plot is that as the distribution around  $Q_{had}^2 = Q_{lep}^2$  is narrow, most radiative photons are soft, ie. low energetic, as noted previously.

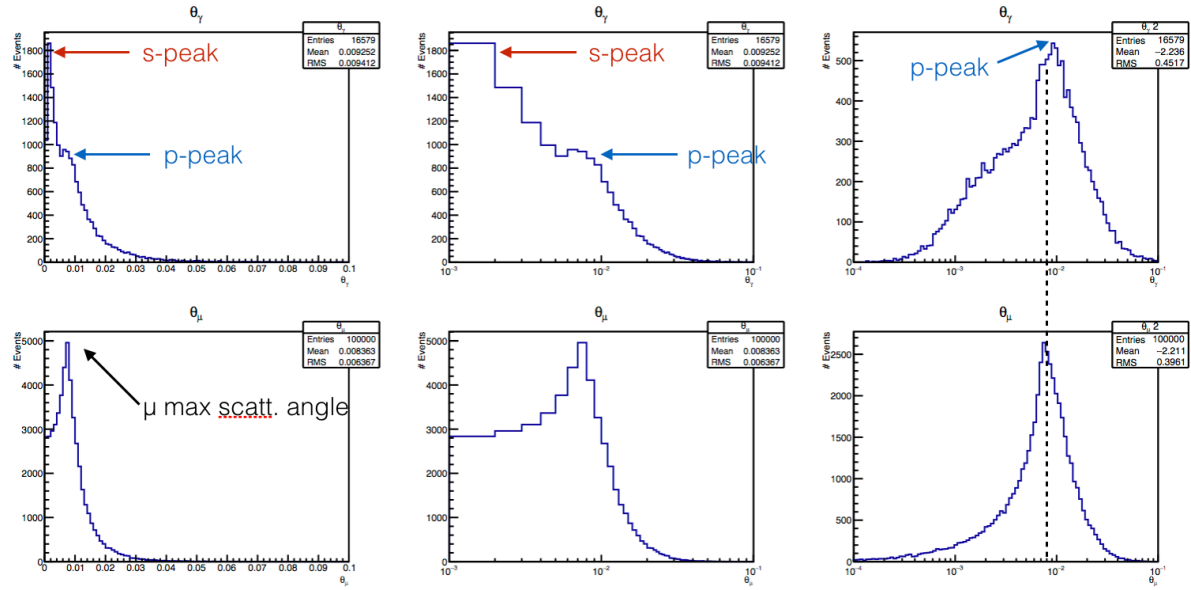


Figure 6:  $\theta$  angle distribution for the radiative photon (up) and the outgoing muon (down). From left to right the scaling of the x-axis is changed (normal, logarithmic, logarithmic with rebinning). One can see two peaks in the theta distribution of the radiative photon : one around zero (s-peak) and one a little bit further (p-peak). When compared to the  $\theta$  distribution of the outgoing muon, especially with the last scaling, one can see that the two peaks match.

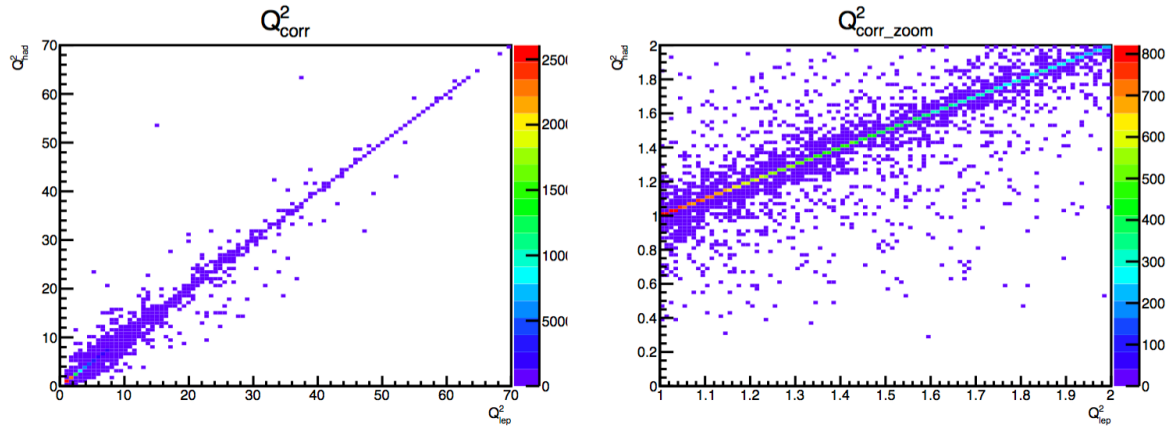


Figure 7:  $Q^2$  correlation plot  $Q^2_{had} = f(Q^2_{lep})$ . On the left is the plot for the complete range of  $Q^2$  given by the kinematic constraints. The scattering around the  $Q^2_{had} = Q^2_{lep}$  line is small, indicating that most of the radiative photons are soft. On the right is the same plot but restricted to the  $Q^2_{lep} \in [1, 2]$  and  $Q^2_{had} \in [0, 2]$  region. The fact is that for  $Q^2_{lep} = 1$ ,  $Q^2_{had}$  takes values in above and below  $Q^2 = 1$ , as expected

## 5 Comparison between DJANGO and RADGEN

With an other radiative event generator used in COMPASS, RADGEN[7], a rather problematic result was obtained : the generator produced a high number of hard photon, leading to great discrepancies with the real data. In Fig. 8, a comparison of DJANGO and RADGEN is shown. The  $E_\gamma$  is compared and one can clearly see that DJANGO is producing less hard photons than soft photons, while this is complete inverse for RADGEN.

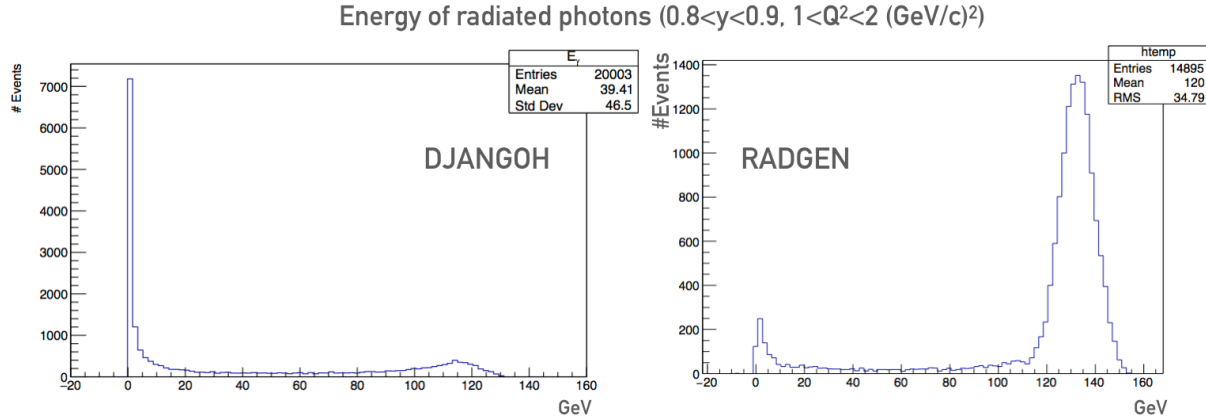


Figure 8: Left is the radiative photon energy distribution for  $0.8 \leq y \leq 0.9$  and  $1 \leq Q^2 \leq 2(\text{GeV}/c)^2$  for DJANGO, right is the same distribution for RADGEN. DJANGO is producing overall more soft photons than hard ones, unlike RADGEN. Nevertheless, the plot does not allow to conclude whether DJANGO is producing less hard photon than RADGEN.

By comparing more thoroughly, computing the proportion of inelastic events in the range between 20 and 160 GeV over the total number of DIS events (Fig. 9), RADGEN has a total of 18.6% of event in this area when DJANGO reaches a total of only 8.6%. We can then definitely claim that DJANGO is producing less hard photons than RADGEN, which is encouraging for the comparison to real data.

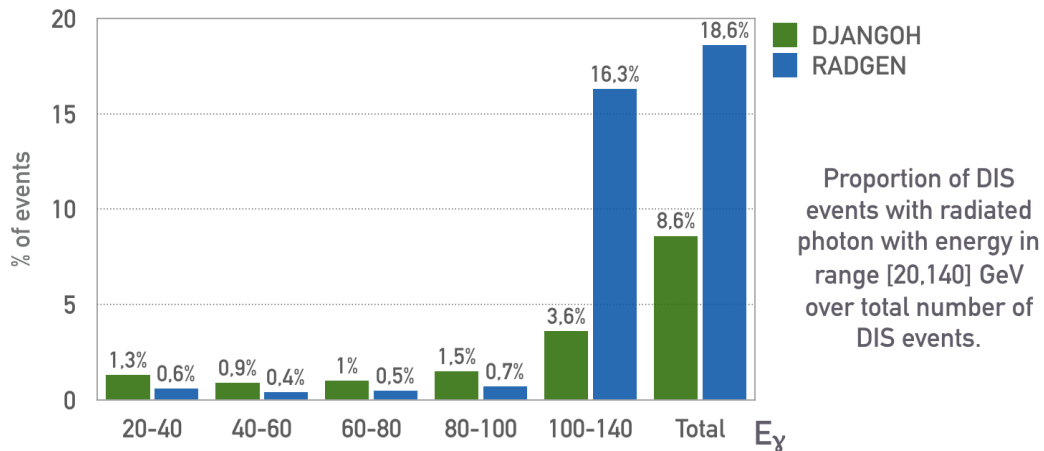


Figure 9: Comparison of proportion of inelastic event in high energy radiative photon range over total number of DIS event with the same kinematical restrictions as in Fig. 8. This comparison allows to conclude that DJANGO is indeed producing less hard photons than RADGEN with more than a factor 2 between the two generators. Thus, RADGEN is producing much less soft photons than DJANGO, when referring to Fig. 8 with this knowledge.



## 6 TGEANT and DJANGO

### 6.1 Forewords : Technical description of DJANGO

The computational procedures applied in DJANGO are based on the methods used in AXO[8] library for Monte-Carlo integration and event generation. AXO relies on the Monte-Carlo integration algorithm VEGAS[9]. The computation is made in this order :

- Integration of the different contributions : partial cross-sections are determined according to the defined phase-space region. They give the relative weight of the corresponding contribution in the final step of event sampling. Moreover, the integration procedure supplies information for the construction of the distribution function applied for an event generation.
- Estimation of the local maxima of the distribution function in a predefined number of hypercubes.
- According to the partial cross-sections that were calculated, events are generated randomly from the individual contributions. HERACLES is only taking care of the scattered lepton and the potential radiative photon. DJANGO is simulating the QCD effect and generates the hadronic part of the event.

An upgrade to the original DJANGO is the possibility to use different input energies for the incoming lepton for mutple event generation. It was before only allowed to specify one input energy at the launch of the program. DJANGO is now capable to take into account a new beam energy at each new event. Nonetheless, using different input energies for event generation is causing a problem : the cross-sections that are needed for event generation are depending on this input energy. The naive way would be to recompute the cross-section for each event but this solution takes much time due to the computation of the cross-section being the slowest part of the generation.

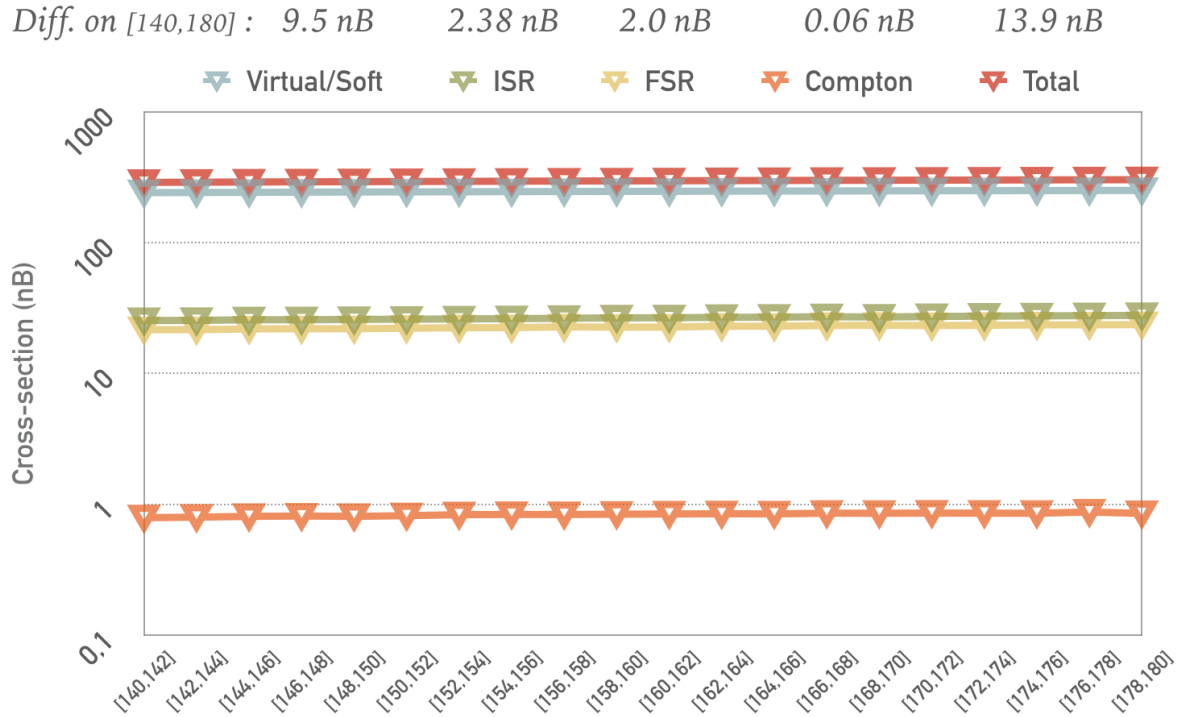


Figure 10: Cross-section values (in nB) for different bins in a range for the energy of the incoming muon between 140 and 180 GeV for Virtual/Soft cross-section (gray blue), Initial State Radiation cross-section (green), Final State Radiation cross-section (yellow), Compton contribution cross-section (orange) and the total cross-section (red). The cross-section distribution is almost flat, with an overall difference between the first and last bin in energy of 13.9 nB for a mean total cross-section of about 500 nB. Though it represents only a 2 percent change in the cross-section, it has to be taken into account for a proper event generation.

A solution to circumvent this problem is the use of a grid of cross-sections. This grid is initialized after a rough specification of the type of dispersion in energy of the considered beam. Basically, the grid needs to have a mean energy and the standard deviation of energy to this mean energy, as well as the number of bins in the grid. Then for each bin, the energy of the center of the bin is taken and the cross-section corresponding to this energy is computed. The narrower the bins are the more accurate the cross-section is for the considered bin. As shown in Fig. 10, with a mean energy of 160 GeV, a distribution width of 20 GeV and 20 bins, the grid is giving an accurate map of the cross-sections. Though the difference of cross-section is not very large (13.9 nB, 2% of the total cross-section), it has to be taken into account for a proper event generation.

The user can communicate with the program by setting initial condition via codewords. Principal useful codewords from the HERACLES part are :

- **EL-BEAM : Quantities defining the properties of the lepton beam.**

**polari** : Polarisation of the beam

**beam** : Particle of the beam ( $e^+, e^-, \mu^+, \mu^-$ )

- **KINEM-CUTS : Definition of the kinematical cuts. These cuts are applied to the leptonic variables  $x_l, y_l, Q_l^2$ , defined by the momentum of the final-state lepton and the mass of the hadron.**

**icut** : Determine the number of active cuts (D=3)

=1 : cuts in  $x_l$  and lower cut in  $Q_l^2$

=2 : cuts in  $x_l$ , lower cut in  $Q_l^2$  and lower cut in  $W_h$ ,

=3 : cuts in  $x_l, y_l, Q_l^2$  and  $W_h$ )

**ixmin** : Minimal  $x_l$

**ixmax** : Maximal  $x_l$

**iymin** : Minimal  $y_l$

**iymax** : Maximal  $y_l$

**iq2min** : Minimal  $Q_l^2$

**iq2max** : Maximal  $Q_l^2$

**iwmin** : Minimal  $W_h$

- **GSW-PARAM : Monitoring the definition of the electroweak parameters and the inclusion of the different virtual corrections.**

**lparin1** : (D=2)

=1 : Electroweak parameters set with fixed values for the boson masses  $M_W, M_Z$ )

=2 : Electroweak parameters calculated from fixed  $M_Z, G_\mu$

**lparin2** : (D=1)

=0 : Only Born cross section without electromagnetic or weak corrections is integrated

=1 : Born cross section including corrections is determined by the values of lparin[3] to lparin[11]

**lparin3** : Flag for inclusion of higher order contributions (D=3)

=0 : No higher order corrections

$\geq 1$  : Terms of  $O(\alpha^2 m_t^4)$  included

$\geq 2$  : Terms of  $O(\alpha \alpha_s m_t^4)$  included

$\geq 3$  : Running  $O(Q^2)$  is used for the radiative cross section

**lparin4** : Leptonic QED corrections (D=1)

**lparin5** : Quarkonic QED correction (D=0)

**lparin6** : Lepton-quark interference (D=0)

**lparin7** : Fermionic contribution to the photon self-energy  $\Sigma^\gamma$  (D=2)

=0 : Not included

=1 : Parametrization with the help of quark masses

=2 : Parametrization from ref 20

**lparin8** : Fermionic contribution to the  $\gamma - Z$  mixing (D=0)

**lparin9** : Fermionic contribution to the self-energy of the Z boson (D=0)

**lparin10** : Fermionic contribution to the self-energy of the W boson (D=0)

**lparin11** : Purely weak contributions to the self-energy, vertex corrections and boxes (D=0)

- **GD-OPT : Definition of the size of the cross-section grid.**

**gdmean** : Central value of the energy distribution

**gdsddv** : Width of the energy distribution

**gdsiz** : Number of bins in the grid

- **EGAM-MIN : Definition of a lower cutoff energy for bremsstrahlung photons (in GeV).**

**egam** : Value of the cutoff

- **INT-OPT-NC : Defines the contributions to the neutral current interactions for which the integrated cross-section is asked to be calculated.**

**inc2** : Integration for the non-radiative contribution (Born term including virtual and soft corrections) (D=1)

**inc31** : Number of iteration for integration of the contribution from initial state leptonic bremsstrahlung by VEGAS (D=18) [NB : There are subtle difference at different scale of iteration (100, 200). These are explicited in DJANGO manual. This remark is valid for all other integration that does iteration.]

**inc32** : Number of iteration for integration of the contribution from final state leptonic bremsstrahlung by VEGAS (D=18)

**inc33** : Number of iteration for integration of the Compton contribution by VEGAS (D=18)

**inc34** : Number of iteration for integration of the contribution from radiation from the quark line by VEGAS (not active in DJANGO, D=0)

**iel2** : Integration of the elastic scattering  $lp \rightarrow lp$  (D=1)

**iel31** : Number of iteration for integration of the quasielastic tail  $lp \rightarrow lp\gamma$  from initial state leptonic bremsstrahlung by VEGAS (D=18)

**iel32** : Number of iteration for integration of the quasielastic tail  $lp \rightarrow lp\gamma$  from final state leptonic bremsstrahlung by VEGAS (D=18)

**iel33** : Number of iteration for integration of the quasielastic tail  $lp \rightarrow lp\gamma$  from the Compton contribution by VEGAS (D=18)

- **INT-POINTS : Upper limit for the number of integration points used by VEGAS (D=10000).**

- **SAM-OPT-NC : Monitors the inclusion of individual contributions to the neutral current cross-section for event sampling.**

**isnc2** : Non-radiative contribution (D=1)

**isnc31** : Initial state leptonic bremsstrahlung (D=1)

**isnc32** : Final state leptonic bremsstrahlung (D=1)

**isnc33** : Compton contribution (D=1)

**isnc34** : Quarkonic radiation (D=0)

**isel2** : Elastic scattering  $lp \rightarrow lp$  (D=1)

**isel31** : Initial state leptonic radiation for  $lp \rightarrow lp\gamma$  (D=1)

**isel32** : Final state leptonic radiation for  $lp \rightarrow lp\gamma$  (D=1)

**isel33** : Compton contribution for  $lp \rightarrow lp\gamma$  (D=1)

- **NUCLEUS : Type and energy of the nucleus.**

**epro** : Energy of the nucleus

**hpolar** : Polarization of the nucleus

**hna** : Number of nucleons in the nucleus

**hnz** : Number of protons in the nucleus

- **STRUCTFUNC : Defines the parametrization of the parton densities or structure functions applied.**

**ilib** : (D=2, icode=10150)

=1 : to choose PDFs from PYTHIA. For more infos on the different values for icode, please refer to DJANGO manual.

=2 : to use PDFLIB[10]. Same remark than previously.

**ilqmod** : The options with  $ilqmod \geq 2$  do not allow event generation (D=0)

=0 : Use unmodified PDFs for all  $Q^2$

=1 : Apply exponentially low  $Q^2$  suppression factor to PDFs.

=2 : For  $Q^2 < 6 \text{ GeV}^2$ ,  $F_1$  and  $F_2$  parametrizations from Brasse.

=3 : For  $Q^2 < 6 \text{ GeV}^2$ ,  $F_1$  and  $F_2$  parametrizations from Abramowicz, Levy, Levin and Maor. Brasse is used for  $W < 2 \text{ GeV}$ .

=4 : For  $x < 0.1$  and  $\nu > 10 \text{ GeV}$ ,  $F_1$  and  $F_2$  parametrizations from Badelek and Kwiecinski. For  $x < 0.1$  and  $\nu < 10 \text{ GeV}$ , Brasse and Stein parametrization is used.

=5 : For  $Q^2 < 10 \text{ GeV}^2$ ,  $F_1$  and  $F_2$  parametrizations from Donnachie-Landschoff together with PDFs.

=6 :  $F_1$ ,  $F_2$  and  $R$  parametrizations from TERAD.

- **FLONG : Inclusion of the longitudinal structure function  $F_L$ .**

**iflopt** = LQCD + LTM\*10 + LHT\*100, where LQCD, LTM, LHT specifies whether QCD contributions, target mass effects and higher twists should be included. See manual of LEPTO 6.5.1 for more details. (D=111)

**parl111** : Accuracy for the integration needed to calculate  $F_L$ . (D=0.01)

**parl19** :  $\kappa^2$ , the scale parameter for higher-twist contributions in  $F_L$ . (D=0.03)

- **ALFAS : Define the value of  $\alpha_s$  needed for the inclusion of  $F_L$ .**

**mst111** : Order of  $\alpha_s$  evaluation (D=1)

=0 : fixed  $\alpha_s$  at the value of parl11

=1 : first-order running  $\alpha_s$

=2 : second-order running  $\alpha_s$

**mst115** : Treatment of the  $\alpha_s$  singularity at  $Q^2 \rightarrow 0$  (D=0)

=0 : allow it to diverge like  $1/\ln(Q^2/\Lambda^2)$

=1 : soften the divergence to  $1/\ln(1 + Q^2/\Lambda^2)$

=2 : freeze  $Q^2$  evolution below  $Q_0^2 = 4 \text{ GeV}^2$

- **NFLAVORS : Minimal and maximal number of flavor to be included in cross-section (DMin=1,DMax=6)**

Principal useful codewords from the DJANGO part are :

- **FRAG : Activation of the fragmentation.**

lst7 : (D=1)

- =-1 : no generation of parton cascades and no fragmentation
- =0 : event generation at the parton level, no hadronization
- =1 : event generation including hadronization and decays of unstable particles

- **CASCADES : Describes the part of the event simulation which is determined by perturbative QCD.**

lst8 : (D=12)

- =0 : no QCD effects, no gluon radiation or boson-gluon fusion
- =1 : including QCD processes (gluon radiation and boson-gluon fusion) according to the first-order matrix elements
- =2 : QCD parton cascade evolution of the initial and final quark
- =3 : QCD parton cascade evolution of the initial quark only
- =4 : QCD parton cascade evolution of the final quark only
- =5 : QCD switched off but target treatment exactly as in parton cascade case
- =9 : simulating QCD cascades in the colour dipole model (as implemented in ARIADNE)
- =12-15 : as 2-5 but parton shower added to the event as obtained from first-order matrix elements

- **MAX-VIRT : Maximal virtuality in parton cascades.**

lst9 : (D=5)

- =1 :  $Q^2$
- =2 :  $W^2$
- =3 :  $W.Q$
- =4 :  $Q^2(1-x)$
- =5 :  $Q^2(1-x)\max(1, \ln(1/x))$
- =6 :  $x_0 W^2$
- =9 :  $W^{4/3}$ , similar as in the colour dipole model

- **SOPHIA : Defines the upper limit for the hadronic mass below which SOPHIA is called for the generation of the hadronic final state (D=1.4)**

## 6.2 Implementing DJANGO in TGEANT

The original DJANGO is a FORTRAN framework. The main problem I encountered is that DJANGO was supposed to be used in the Monte-Carlo simulation of the COMPASS apparatus, TGEANT. As the whole simulation is coded in C++, the inclusion of a new generator inside the chain is done via the integration of a plugin in C++ containing the generator. Thus, in order to include DJANGO into TGEANT, I had to modify DJANGO so that it can be used as a C++ plugin generator by TGEANT.

Fig. 11 shows the principal way of event generation in TGEANT.

First, a particle is selected with the T4PrimaryGen class, that reads a given beamfile, and pick a particle at random. Then the T4ProcessBackEnd class generates the given process and creates the outgoing particles that will be detected afterwards by the detectors. This T4ProcessBackEnd class is purely virtual and is replaced by every generator class implemented in TGEANT.

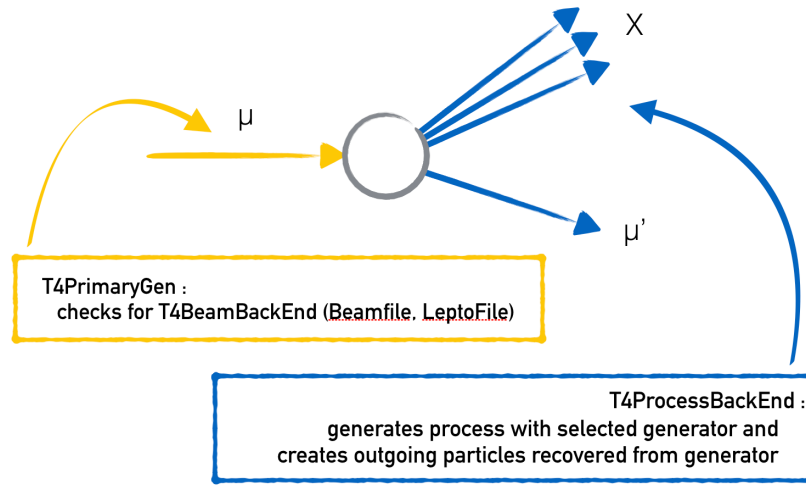


Figure 11: Sketch showing how event generation is handled in TGEANT. A particle is first chosen in the beamfile, then is sent to the dedicated generator that generates the given process. TGEANT then recovers the list of particles from the generator, creates and propagates them.

There are two ways to implement a generator in TGEANT :

- As an internal generator (Pythia and HEPGEN++ way) : a C++ interface to the generator is needed, the conservation of  $P(p, E)$  is perfect and it only needs a beamfile for the primary generator.
- As an external generator (Lepto) : the beamfile is read by the standalone generator, the primary generation and process infos are stored inside a file and TGEANT has to do an extrapolation to -9 meters as the primary generation was done outside of it, occasioning  $P(p, E)$  being not perfectly conserved.

The external implementation is quite complicated as three classes are needed in order to use the generator in TGEANT. First, the beamfile has to be read by the generator which then outputs a file containing the beamfile infos but in its own format. Then a first class has to be dedicated to the reading of the beamfile, recovering the information about the selected muon in the file. A second class takes care of passing of information between the generator and TGEANT. The last class is the class of the generator itself (Fig. 12).

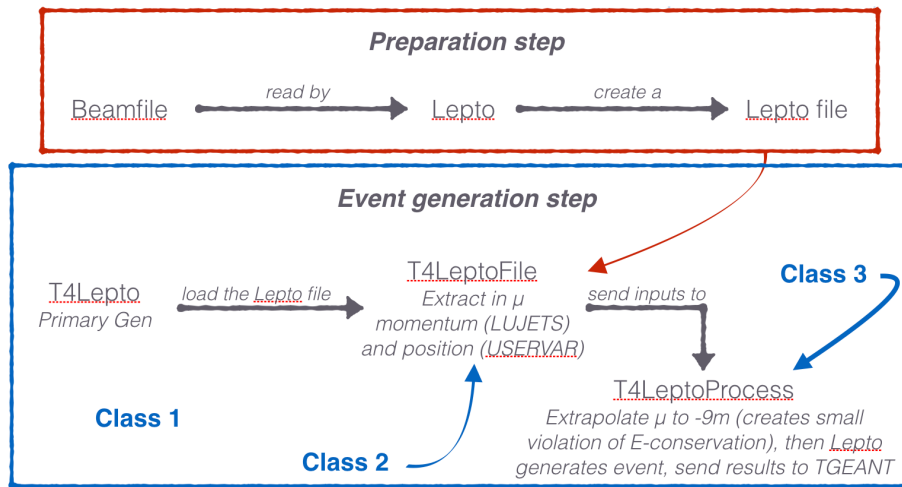


Figure 12: Diagram explaining the philosophy of the external generator implementation, taking the implementation of LEPTO as an example. First and foremost, a specific beamfile is created after the initial one. The external generator will read this pregenerated file to extract infos about the incoming particle. As these informations are at the interaction point, a backward propagation extrapolation in the target material has to be made, inducing some small violations of energy conservation. Then the generator is producing the event and sends the results to TGEANT.

If you apply this method to DJANGO, then you encounter several critical problems :

- A new file format for the converted beamfile has to be created.
- DJANGO has to be modified to allow backward propagation of the incoming muon.
- The result of fragmentation (LUJETS) has to be recovered in a file.

This results in many file accesses and consequently it is a non-efficient way to implement DJANGO inside TGEANT.

The internal implementation requires perhaps more work on the generator itself however this solution is much more efficient. Here, only two class are needed. One is the interface class that creates instances of DJANGO that can be manipulated in any C++ environment. This class is a C++ interface that is handling the FORTRAN part of DJANGO. The other class is taking care of passing the information between the interface to the generator and TGEANT (Fig. 13).

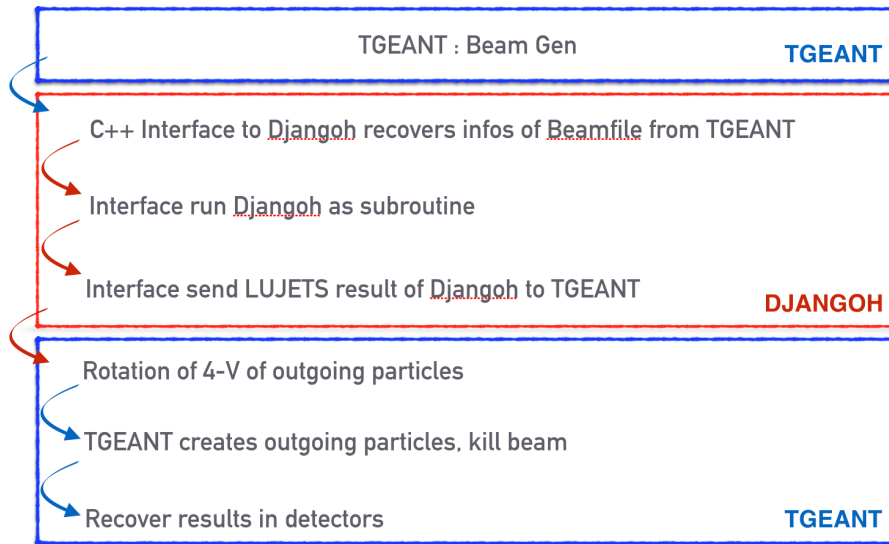


Figure 13: Diagram explaining the philosophy of the internal generator implementation. The beamfile is read by TGEANT and the C++ interface to the generator is recovering the informations of the incoming particle. This interface then runs the generator as a subroutine and sends back the results of the generation to TGEANT, which then creates the outgoing particles accordingly.

This second method was used to implement DJANGO inside TGEANT.

### 6.3 TDJANGO : a C++ interface to DJANGO

In order to create a C++ class (called TDJANGO) that plays the role of an interface, I had to modify some FORTRAN parts of DJANGO, especially the input method. DJANGO is working with an input file where codewords with set values are specified in order to configure the generator. This was not convenient for the idea of an interface. Thus I have drawn correspondances between Common Blocks in FORTRAN and structures in C++ so that I can specify values in the C++ structure and the change is repercutated in the FORTRAN code and vice-versa. It is useful to specify the values for the input but also to recover the results of the hadronization that are located in the LUJETS Common Block. The idea is that within TGEANT the user specifies the input for DJANGO, the interface pass it to the generator and the interface recovers the results of the generator and gives it to TGEANT.

The principal and most useful methods that are included in the interface are :

- **static TDjangoh \*Instance()** : Allow the creation of an instance of TDJANGO. Only one instance at a time is permitted.
- **void ReadXMLFile(const string pFilename)** : Read an XML file in order to initialize the input parameters of DJANGO.
- **void Initialize()** : Initialize the generator with previously defined values and computes cross-sections.
- **void GenerateEvent()** : Generates an event. Always use after having called Initialize() !

Some unavoidable toolbox methods are also present :

- **void SetParticle(const char\* pname)** : Modify the beam particle (useful to switch between  $\mu^+$  and  $\mu^-$ )
- **void Configure(float beam\_e, float pol)** : Modify the beam properties (energy, polarisation)
- **void BornWOqelNC()** : Call for event generation only taking into account the Born cross-section without quasi-elastic.
- **void RClepWOqelNC()** : Call for event generation taking into account the Born cross-section + ISR,FSR and Compton, without quasi-elastic.
- **void EndRecap()** : Print a recap in TDjangoh\_out.dat at the end of the generation.
- **void Clean\_File()** : Clean all the .dat files created by the generator.

For informations about the other methods that are available within the interface, please refer to the doxyfile of TDJANGO.



## 7 TDJANGO : How to ?

In order to get TDJANGO running, recover a version in the following git repository :

```
https://github.com/nipierre/TDJANGO.git
```

You need also a Fortran version of LHAPDF (version below LHAPDF 6.x.x). Then you have to export some environment variables :

```
export TDJANGO=/path/to/TDJANGO
export LHAPDF5=/path/to/LHAPDF5
```

You can go to the root of TDJANGO package and do the following :

```
make setup
make
```

In the test programs :

- test allows you to do full event generation. You have some options if you enter some flags, do bin/test -h to have more infos about it.
- xsen allows you to compute cross-section rapidly, without any hadronisation in order to gain time. Do bin/xsen -h for more infos.

In the utilities :

- rc\_calc allows you to plot the inclusive radiative corrections after different kind of inputs. The most convenient one is to use the results from xsen and feed it directly with the -sigf option.
- sirc allows you to plot the multiplicities for Born and Born+ $\mathcal{O}(\alpha)$  and see the comparison between the two. The distribution are in  $(x_{Bj}, y, z)$  and  $(x_{Bj}, y, p_T)$
- pdist allows you to plot various information about the radiative photons, like the energy distribution, angle distribution, etc.
- lowQ2 is a test code for plotting 3D plots of  $F_2$  functions in  $(x, Q^2)$ . Its usefulness is limited, but can be informative when comparing different parametrization of  $F_2$ .

The XML configuration files are located in the utils/ directory.

For using TGEANT with DJANGO, recover a version of TGEANT supporting DJANGO (still in dev branch ATM but will be merged with master eventually). You can use the script in the directory TGEANT\_script/ (after copying the scripts at the root of where you want to install TGEANT):

```
./fetch_repo.sh
```

In order to compile TGEANT, PYTHIA support needs to be deactivated as the version of Pythia inside DJANGO is creating conflict when it is called within TGEANT and I found no way to circumvent properly this problem. You can use the script in the directory TGEANT\_script/ :

```
./build_TGEANT.sh 1
```

\$CLHEP should be defined and is the root location of your CLHEP package.

NB : \$CLHEP should be /afs/cern.ch/compass/tgeant/TGEANT\_REFERENCE/deps/clhep/install/ .

Afterwards, all you have to do is to select the DJANGO generator in the interface and the djangohSettings\_dvcs2016.xml file from \$DJANGO/utils will be read for configuration of DJANGO.

## 8 Radiative correction factors

### 8.1 Inclusive radiative correction factors

The calculation of the inclusive radiative correction factors can be done by computing  $\sigma_{Born}$  and  $\sigma_{Born+o(\alpha)}$ . A correct way to obtain these factors is the following :

$$\eta(x, y) = \frac{\sigma_{Born}(x, y)}{\sigma_{Born+o(\alpha)}(x, y)} = \frac{\frac{\sigma_{Born,tot} * N_{Born}(x, y)}{N_{Born,tot}}}{\frac{\sigma_{Born+o(\alpha),tot} * N_{Born+o(\alpha)}(x, y)}{N_{Born+o(\alpha),tot}}}$$

The results that are presented below are obtained with the TERAD  $F_2$  and R parametrizations, which describes accurately the behaviour of  $F_2$  at low  $Q^2$ [11] :

- $F_2^p(x, Q^2)$  for  $Q^2 > 0.2 \text{ GeV}^2$  and  $0.000035 < x < 0.85$ , as obtained from a fit to the world proton (and deuteron) data made by the SMC[12].
- For  $Q^2 < 0.2 \text{ GeV}^2$ , a phenomenological model of Badelek and Kwiecinski[13], valid at  $10^5 < x < 0.1$  and  $0 < Q^2 < 1000 \text{ GeV}^2$ .
- $R(x, Q^2)$  as parameterised by SLAC (newer version, called R1998[14]), valid for  $Q^2 > 0.5 \text{ GeV}^2$ , extended to lower values of  $Q^2$ , including the  $R \simeq Q^2$  behaviour at  $Q^2 = 0$ .

The  $o(\alpha)$  are included except for quark line radiation. The reason why these corrections are not included is that these corrections are negligible except at large  $x \geq 0.5$  and  $Q^2 \geq 10^3 \text{ GeV}^2$ , where the corrections reach the magnitude of barely one percent (see Fig. 14). Also, even though this is not the case in TERAD  $F_2$ , these corrections are often not subtracted inside the parametrization, thus they are already taken into account in the parametrization[15].

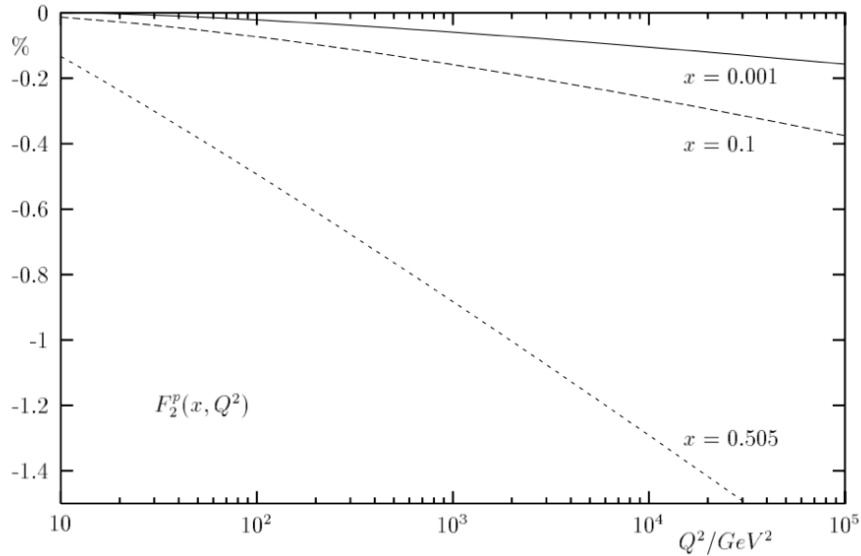


Figure 14:  $Q^2$  dependence of the quarkonic QED corrections (in percent) to the structure function  $F_p^2$  for deep inelastic lepton-proton scattering at  $x = 0.001$ ,  $x = 0.1$  and  $x = 0.505$ .

The kinematic range for the presented results is  $x \in [0.004, 0.9]$  and  $y \in [0.1, 0.9]$ .

In the next figures, DJANGO results are compared with TERAD[16]. The two programs are using the same set of  $F_2^p$  and  $R$ . This are the only inputs (apart from the process input ie.  $\mu p$  scattering at 160 GeV muon energy) that need to be identical so that the comparison is relevant. One thing to be noted is that TERAD is using more complete sets of correction. They should have an impact on the cross-section of TERAD, but are negligible.

The first check for consistency is to compare the  $\sigma_{Born}$  of both programs. Using the same input information on  $F_2^p(x, Q^2)$  and  $R(x, Q^2)$  does not guarantee that  $\sigma_{Born}$  for both program, called  $\sigma_{Born}^D$  and  $\sigma_{Born}^T$ , is the same : the reason is that for example in TERAD,  $\sigma_{Born}^T$  is computed without constants like  $\pi$ ,  $M_{proton}$ ,  $\alpha$ , etc. and its functional form is unknown, unlike in DJANGO. This means that the ratio  $r = \frac{\sigma_{Born}^D}{\sigma_{Born}^T}$  may differ from 1 but must be constant as a function of  $x$  and  $y$ . Fig. 15 shows that this is the case.

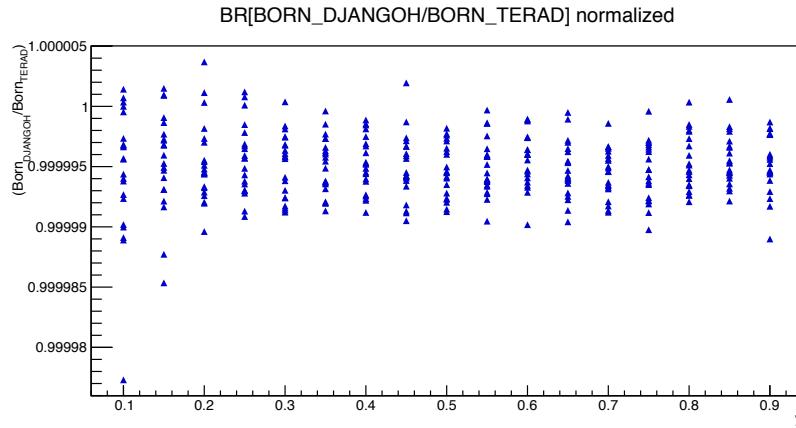


Figure 15: Ratio of the Born cross-sections calculated with DJANGO and TERAD, for the same input, as a function of  $y$  at different values of  $x$  (staggered points at fixed  $y$ )

The radiative correction factors  $\eta(x, y)$  for DJANGO and TERAD are compared in Fig. 16. The absolute difference is then plotted in Fig. 17, showing that the two programs differ at most 3% in the region of lowest  $x$  and highest  $y$ .

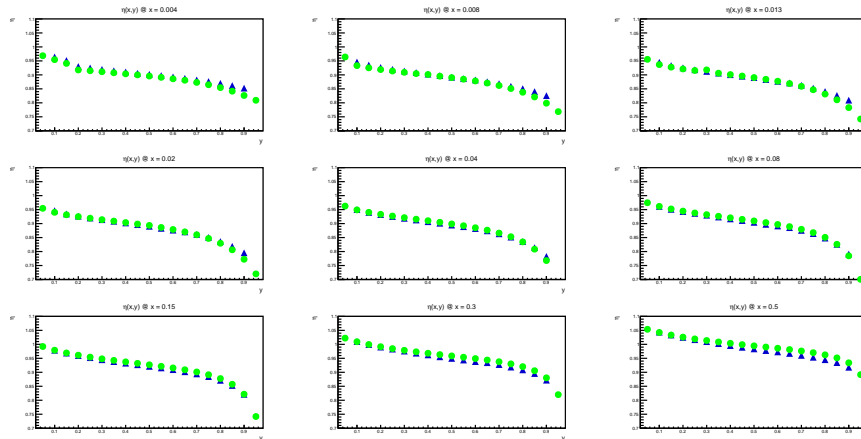


Figure 16: Comparison of radiative corrections factor  $\eta(y)$  for fixed values of  $x$ , computed for proton target at 160 GeV and with the input information given above. Green dots mark results of TERAD, blue triangles the results of DJANGO.

Overall, the corrections on the considered kinematic space are within 10% except for high  $y$  corrections that go up to 40%.

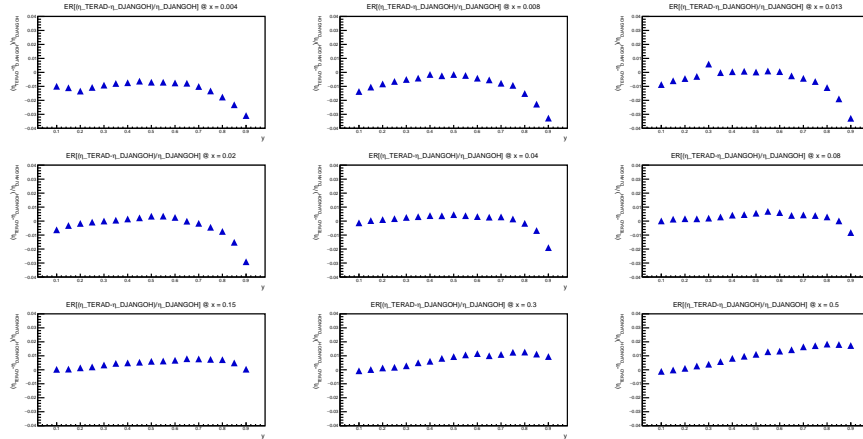


Figure 17: Ratio of radiative corrections factors obtained in Fig. 16,  $(\eta_T/\eta_D) - 1$  as a function of  $y$  for fixed values of  $x$

## 8.2 Semi-Inclusive radiative correction factors : Effect on Multiplicities

The calculation of the semi-inclusive radiative correction factors can be done by computing  $M_{Born}^{h^\pm}$ , multiplicities obtained without radiative corrections, and  $M_{Born+o(\alpha)}^{h^\pm}$ , multiplicities obtained with radiative corrections. A correct way to obtain these factors is the following :

$$\eta^{h^\pm}(x, y, z) = \frac{M_{Born}^{h^\pm}(x, y, z)}{M_{Born+o(\alpha)}^{h^\pm}(x, y, z)}$$

In the following plots, the cuts from the SIDIS analysis for the selection of DIS events and hadrons are used. For more information, please refer to the multiplicities papers[17]. I will explicitate the kinematical cuts that are used :

- $0.004 \leq x_{Bj} \leq 0.4, x_{Bj} \in \{.004, .01, .02, .03, .04, .06, .1, .14, .18, .4\}$
- $0.1 \leq y \leq 0.7, y \in \{.1, .15, .2, .3, .5, .\}$
- $3 \leq p_h \leq 40 \text{ GeV}$

Fig. 18 exhibits the semi-inclusive radiative correction factor  $\eta^{h^\pm}(x, y, z)$ . This factor goes from 0% correction at low  $z$  and low  $y$  to 10% correction at high  $z$  and high  $y$ . This dependance on  $y$  and  $z$  is expected (eg. if a hadron has a high  $z$  in a non-radiative event, consider the same event but with the radiation of a real photon,  $\nu_{lep}$  will remain the same but the hadron will have in reality less energy available from the virtual photon, thus having  $z_{had} \leq z_{lep}$ , leading to less events in the high  $z$  region for the multiplicities obtained with radiative corrections).

Another view on the semi-inclusive correction is given by Fig. 19, where the  $z$  binning is replaced with a  $p_T$  ( $\eta^{h^\pm}(x, y, p_T)$ ). The factor goes from -20% correction at high  $p_T$ , high  $y$  and high  $x$  to +10% correction at low  $y$ , high  $x$  and high  $p_T$ .

When looking at  $\eta^{\pi^\pm}(x, y, z)$  and  $\eta^{K^\pm}(x, y, z)$ , the corrections are very similar to  $\eta^{h^\pm}(x, y, z)$ .

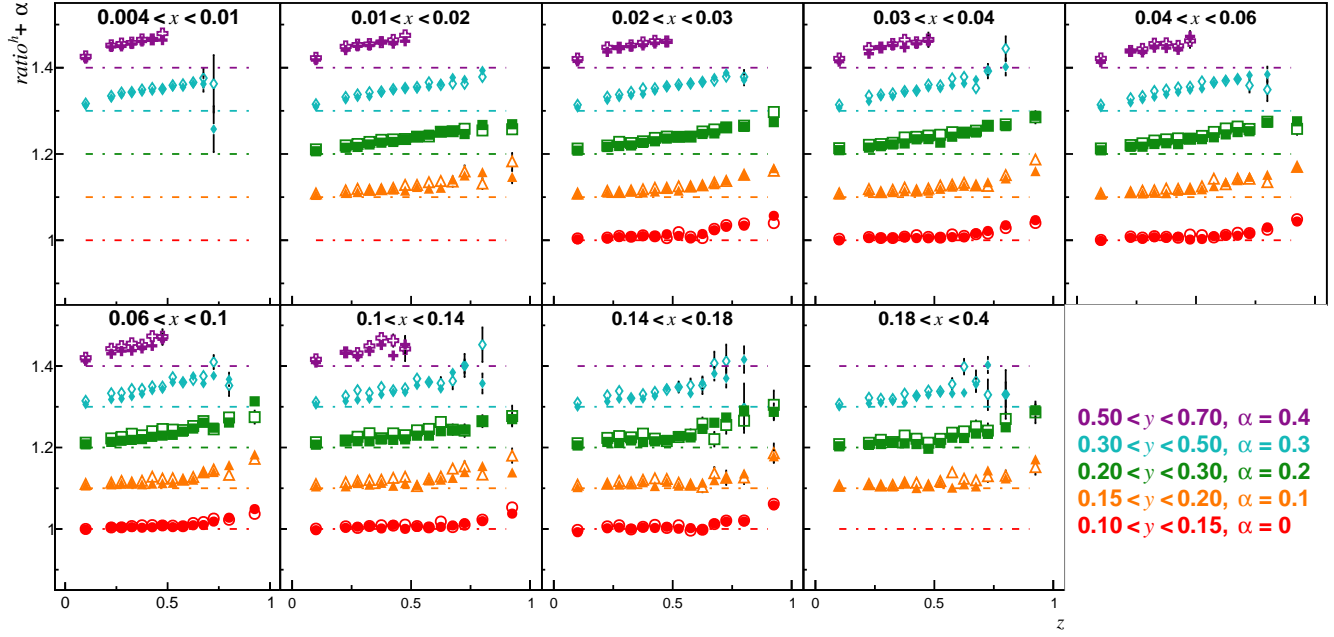


Figure 18:  $\eta^{h\pm}(x, y, z)$ , positive hadrons in full points, negative in voided points, in bins of  $x$ , staggered with  $y$  and versus  $z$ . The corrections go from 0% at low  $z$  and low  $y$  to 10% at high  $z$  and high  $y$

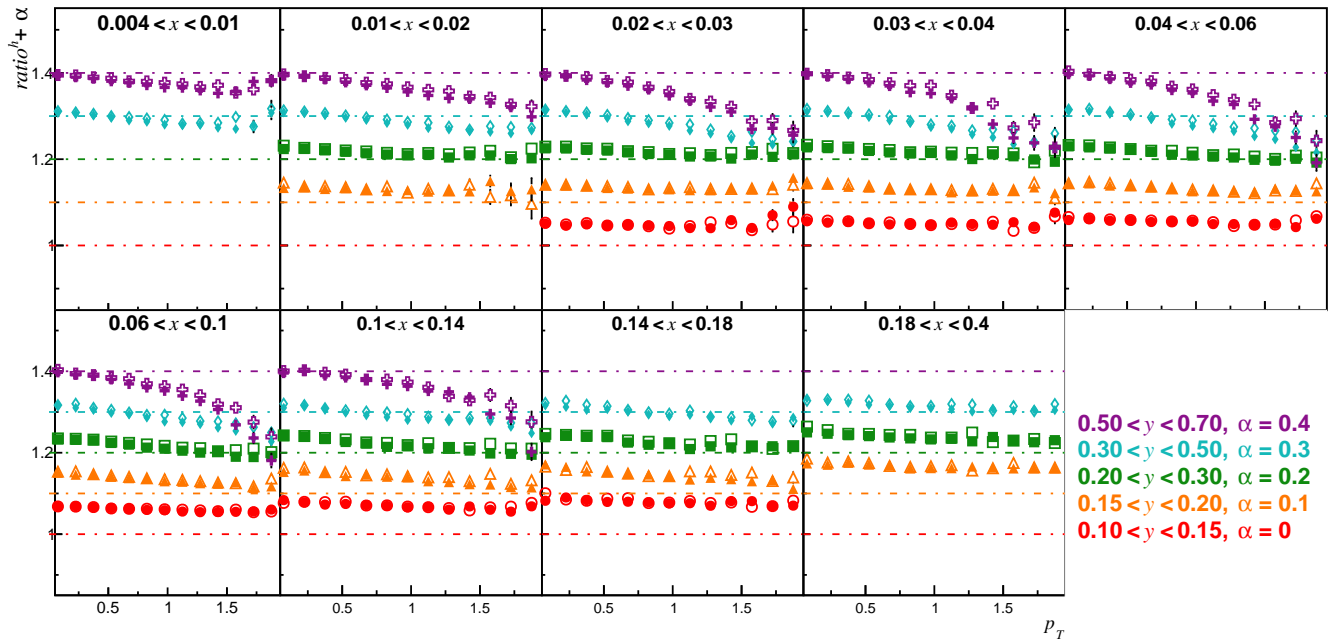


Figure 19:  $\eta^{h\pm}(x, y, p_T)$ , positive hadrons in full points, negative in open points, in bins of  $x$ , staggered with  $y$  and versus  $p_T$ . The corrections go from -20% at high  $p_T$ , high  $y$  and high  $x$  to +10% at low  $y$ , high  $x$  and high  $p_T$ .

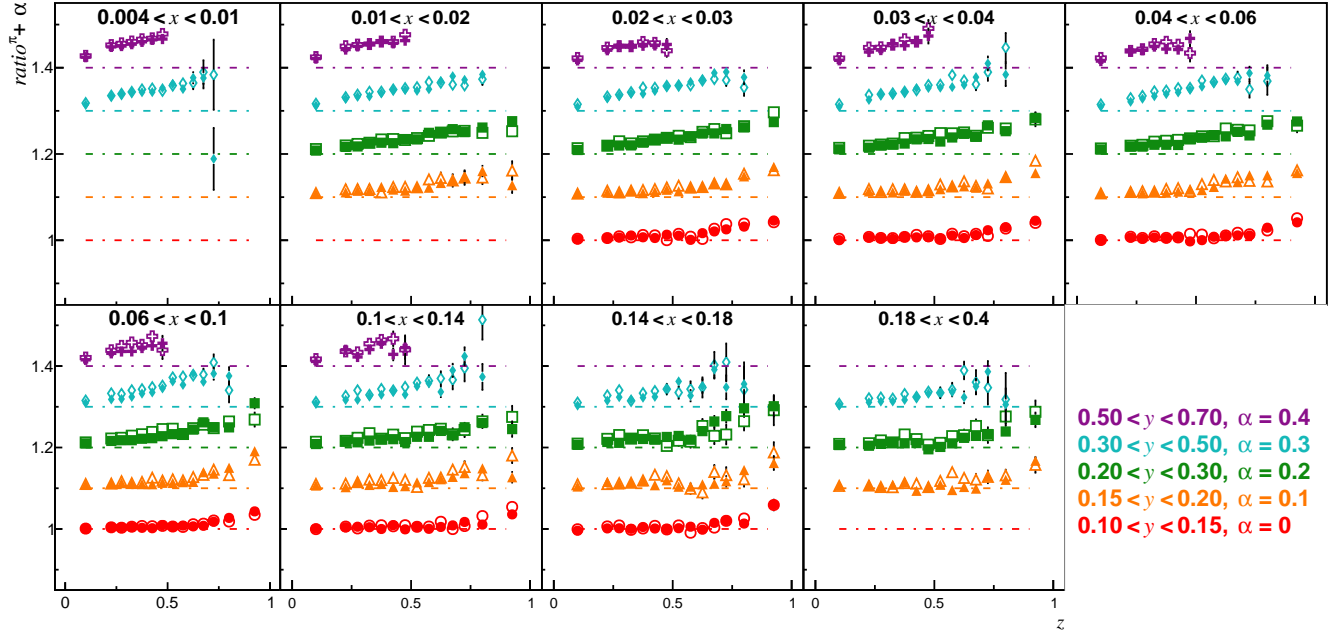


Figure 20:  $\eta^{\pi^{\pm}}(x, y, z)$ , positive hadrons in full points, negative in open points, in bins of  $x$ , staggered with  $y$  and versus  $z$ . Same conclusions as in Fig. 18 can be made.

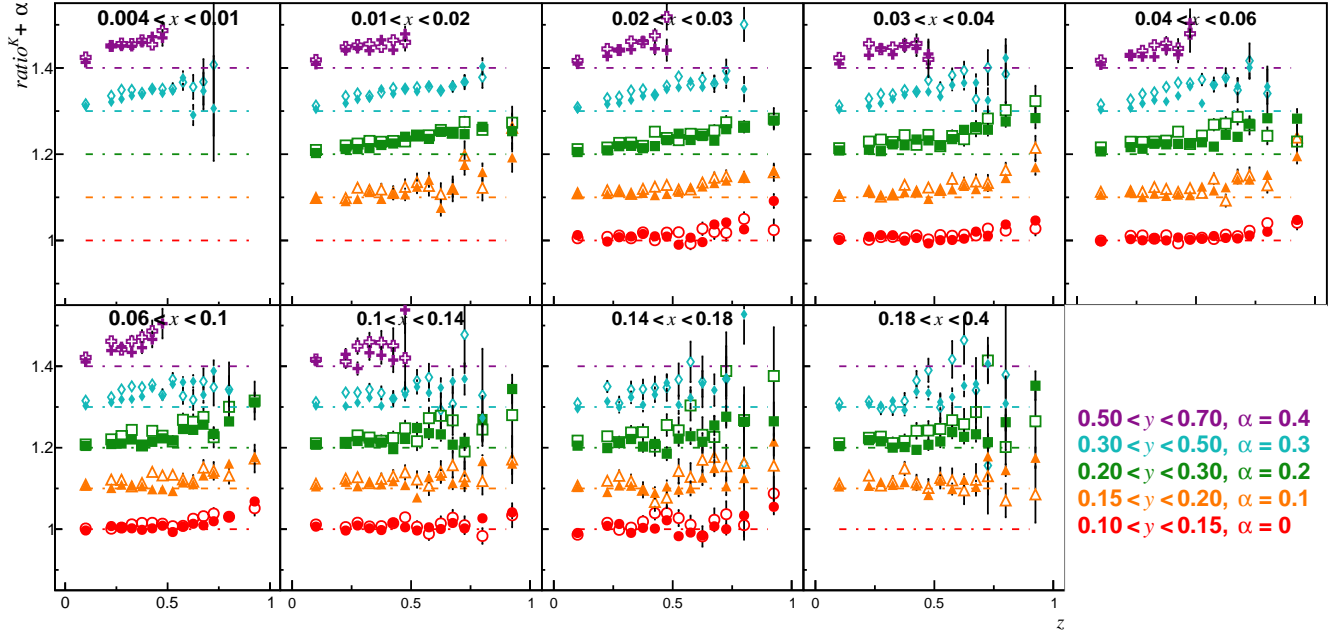


Figure 21:  $\eta^{K^{\pm}}(x, y, z)$ , positive hadrons in full points, negative in open points, in bins of  $x$ , staggered with  $y$  and versus  $z$ . Same conclusions as in Fig. 18 can be made.

## 9 Conclusion

The program DJANGO produces the same inclusive radiative corrections  $\eta(x, y)$  as TERAD when the same input structure functions are used. In the kinematic range of COMPASS, the differences are at most 3% of the hadronic correction value  $\eta(x, y)$ . Most probably they are due to rounding errors (more serious in TERAD) and also to excluding certain processes (e.g. radiation from quark lines) in DJANGO. The results on the semi-inclusive corrections  $\eta(x, y, z)$  seem coherent with what is expected ie. a mean correction of 5% on hadrons increasing with  $y$  and  $z$ .

DJANGO can also now be used easily as a generator inside TGEANT thanks to the TDJANGO interface in order to produce Monte-Carlo with radiative corrections included.

## 10 Acknowledgement

I am deeply grateful to Hubert Spiesberger from University of Mainz for guiding me through DJANGO and for numerous enlightening discussions on radiative corrections.

## References

- [1] E. C. Aschenauer, T. Burton, T. Martini, H. Spiesberger, M. Stratmann, Phys. Rev. D **88** (2013) 114025.  
H. Spiesberger, HERACLES and DJANGO : Event Generation of  $ep$  Interactions at HERA Including Radiative Processes (Version 1.6) (2005, available on : <http://wwwthep.physik.uni-mainz.de/~hspiesb/djangoh/djangoh.html>) and reference therein, especially [9] and [13].
- [2] G. Ingelman, A. Edin, J. Rathsmann, LEPTO 6.5: A Monte Carlo generator for deep inelastic lepton - nucleon scattering, Comput.Phys.Comm. **101** (1997) 108-134
- [3] L. Lönnblad Comput. Phys. Commun. **71** (1992) 15.
- [4] T. Sjöstrand, S. Mrenna, and P. Z. Skands, JHEP 05, 026 (2006), hep-ph/0603175.
- [5] T. Sjöstrand, Comp. Phys. Commun. **39** (1986) 347.  
T. Sjöstrand and M. Bengtsson, Comp. Phys. Commun. **43** (1987) 367.  
T. Sjöstrand, CERN-TH.6488/92.
- [6] A. Mücke, R. Engel, J. P. Rachen, R. J. Protheroe and T. Stanev, Comput. Phys. Commun. **124** (2000) 290.
- [7] I. Akushevich et al., RADGEN 1.0: Monte Carlo generator for radiative events in DIS on polarized and unpolarized targets, In \*Hamburg 1998/1999, Monte Carlo generators for HERA physics\* 554-565 hep-ph/9906408.
- [8] S. de Jong, J. Vermaseren, AXO User Manual, NIKHEF-H Report (1987).
- [9] G. P. Lepage, J. Comp. Physics **27** (1978) 192.
- [10] G. Altarelli, R. Kleiss and C. Verzegnassi (Eds.), Z Physics at LEP 1, Vol. 3: Event generators and software, CERN 89-08, 21 September 1989.  
B. Bambah et al., QCD generators for LEP, CERN-TH. 5466/89 (1989).
- [11] N. Pierre, B. Badelek, Comparison of radiative processes calculations in TERAD and DJANGO, COMPASS note 2017-3.
- [12] SM Collaboration, B. Adeva et al., Phys. Rev. D **58** (1998) 112001 (Appendix).
- [13] J. Kwiecinski and B. Badelek, Z. Phys. C **43** (1989) 251; B. Badelek and J. Kwiecinski, Phys. Lett. B 295 (1992) 263.
- [14] E143 Collaboration, K. Abe et al., Phys. Lett. B **452** (1999) 194; see also the Appendix in COMPASS Collaboration, V.Yu. Alexakhin et al., Phys. Lett. B **647** (2007) 33.
- [15] H. Spiesberger, Phys. Rev. D **52** (1995) 4936-4940.
- [16] A.A. Akhundov, et al., Fortsch. Phys. **44** (1996) 373.  
B. Badelek, D. Bardin, K. Kurek and C. Scholz, Z. Phys. C **66** (1995) 591.  
B. Badelek, TERAD15 user guide, COMPASS note 2015-6. **This note is still under construction!**
- [17] COMPASS Collaboration, C. Adolph et al., Phys. Lett. B, **764** (2017), 1-10  
COMPASS Collaboration, C. Adolph et al., Phys. Lett. B, **767** (2017), 133-141