Tutorial of Mathematica Program

Bo Ting Wang[1]

[1] *Department of Physics, Southern Methodist University, Dallas, TX 75275-0181, USA*

(Dated: November 2, 2017)

Abstract

No

Contents

I Physics Part

II Mathematica Program

Part I
Physics Part

I. A SYSTEMATIC APPOACH TO QUANTIFY CONSTRAINTS ON PDFS IMPOSED BY EXPERIMENTAL DATA

A. Introduction

1. Situation

   Parton distribution functions (PDFs) are crucial for understanding the behaviour of hadron collisions and then exploring the Standard Model (SM). PDFs describe the structure of hadrons, which affect the configurations of the final particles in the collisions. Therefore, the magnitudes of physical observables in hadron collisions strongly depend on PDFs. Currently, The Large Hadron Collider (LHC) produces a lot of experimental data. Owning to the fact that uncertainties in measurements constantly decrease, reducing the PDF uncertainties of physical observables and using the higher order PDFs will make it easier to find the inconsistency between SM and the data sets collected by the LHC and then discover new physics. Incorporating more (LHC) new data sets in the global fits of PDFs is a naive way to generate better PDF sets with small uncertainties.

2. Problem Statement

   However, incorporating more experimental data points will substantially increase the time for fitting PDF sets, especially when we fit higher order PDF sets. From here we know that how to select data sets in global fits will become extremely important in the near future. It is essential to know which data sets will effective constrain the higher order PDFs for the global fits in the limited time of computation. In addition, because physical predictions are sensitive to respective flavours and regions of $\{\xi, \mu\}$ in PDFs, we need to narrow down uncertainties of the specific regions of $\{\xi, \mu\}$ (in the PDFs). Where partonic ξ are momentum fractions and μ are QCD factorization scales. For example, if PDF values for the leading $\{\xi, \mu\}$ ranges and flavours that characterize kinematical quantities for Higgs production processes (e.g. at $\mu = 125\,GeV$) are tightly constrained, the theoretical predictions for these processes are reliable (precise).

   (a) methods

      Using correlation between PDF uncertainties in two observables have been proposed to study constraints on PDFs and constraints on observables imposed by PDFs [?][?][?].

   (b) the evaluation of the methods

      The approach can help us to find the $\{\xi, \mu\}$ ranges of PDFs affecting physical observables such as total cross section [?]. It is yet be established that how to know the ranges specifying PDFs constrained by experimental data sets.

3. Objectives

   Thus, establishing a better understanding of the relationships between the strength of constraints on PDF and experimental data sets will be a significant and beneficial contribution to particle physics.

4. Methodology

   I have developed and tested a systematic method to study the constraints on PDFs imposed by the experimental data sets. I will use established statistical observables to quantify the strength of these constraints. After that, I will introduce a statistical technique to visualize the regions of partonic momentum fractions ξ and QCD factorization scales μ where the experiments impose strong constraints on the PDFs. Recent experimental data will be considered in the analysis in order to provide better constraints to various ranges of PDFs.

5. Scope, Limitations and Asumptions

6. Significance

7. Structure of This Paper

   The article proceeds as follows. First, in the I B 1, we give a brief overview of PDF fitting. Second, the method used to "see" PDF from experimental data is introduced in I B 2. Third, an idea to estimate constraint of PDFs that we have seen from this method is provided in I B 3. Fourth, advantages of using correlation, residual uncertainty, and sensitivity to study PDF constraint is discussed in I C 1. Two methods are discussed in I C 2 and I C 3. The corresponding Mathematica code is introduced in II.

B. Overview of Global fitting and Constraints on PDFs

...

1. Introduction of PDF fitting

To fit PDF sets, we first need to determine the input theoretical model and data sets. The model includes the selection of quark mass, coupling constants and the order considered in the correction of perturbation (i.e. LO, NLO and etc). Then we determine the parametrization function form $f_q(\xi, Q_0) = a_0\xi^{a_1}(1-\xi)^{a_2}F(a_3, a_4, ...)$ at the lowest factorization scale Q_0, for which we need to take some physical rules into account. For instance, $a_0\xi^{a_1}(1-\xi)^{a_2}$ term requires that the probabilities of the partons with momentum fraction $\xi = 1$ or $\xi = 0$ are 0. Besides, the momentum conservation ($\int_0^1 \xi \sum_i f_i(\xi, Q)dx = 1$) requires that the total momentum of all subparticles in each hadron should equal to the kinematical momentum of that hadron. ($\int_0^1 (u(x) - \bar{u}(x))dx = 2$ and $\int_0^1 (d(x) - \bar{d}(x))dx = 1$) require that protons consist of uud at the low factorization scale. we use χ^2 minimization to explore the best fit parameters a_0, a_1, a_2 and etc. χ^2 analysis applies the ratios of the deviations between theoretical predictions and experimental values to experimental error bars to quantify the goodness of fits. Here are steps of the PDF fitting:

1. select the experimental data sets and theoretical model in global fits
2. write down parametrization functions of all flavours
3. determine the best-fitted a_0, a_1, a_2...... of the parametrization functions by minimizing χ^2
4. determine uncertainties of the parametrization functions by requiring $\chi^2 < \chi^2_{min} + \chi^2_{torelance}$

2. PDF fitting with χ^2 Method

χ^2 test is a way to evaluate the goodness of fits. We assume good fits are theoretical predictions within experimental error bar. Thus, by residuals (r_i) of data points i, We can estimate the goodness of the fit of this point. For data sets, the goodness of fits is the sum of all points i and r_i, which means that we use the squared distance to evaluate the agreement between the model and the data sets. Hence, we can obtain the best coefficients of the model by minimizing χ^2. If $\chi^2 \gg N_{data}$ for a data set and a theory, this fit is bad. If $\chi^2 \leq N_{data}$, this fit is better than expected. If $\chi^2 \ll N_{data}$, this fit is overfitted. To sum up, r_i and χ^2 could be defined as follows:

$$r_i = \frac{T_i - D_i}{\sigma_i}$$

$$\chi^2 = \sum_{i \in \text{Expt data}} r_i^2$$

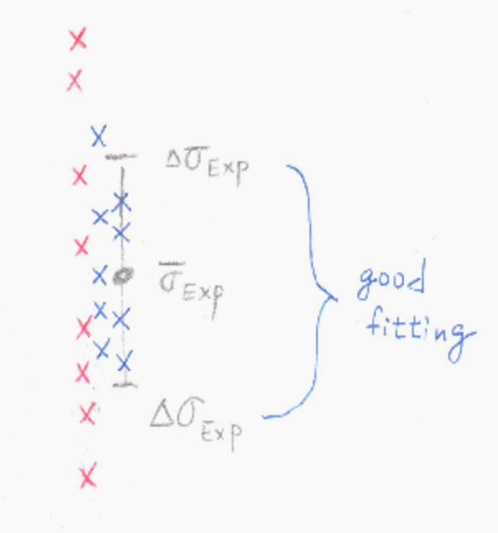
Where i r_i, T_i, D_i, and σ_i are the residual, theoretical prediction, experimental central value, and experimental uncertainty in data point i. When we take systematic uncertainties into account, the experimental central values should be modified to $D_{shift,i} = D_i + shift_i$ since the averages of the measurements are shifted from the real values. Here we provide criteria of good PDF fittings

1. $\chi^2 \simeq N_{data}$, the smaller a χ^2, the better a fitting
2. if residuals of points i are small ($|r| < 1$, r is called residual $\frac{T_i - D_{shift,i}}{\sigma_i}$), the fit of these points is good

3. How to estimate PDF constraints

In general, we estimate the uncertainties of the parameters describing PDFs by constraining χ^2 value. In other words, we identify the region representing to the parameter uncertainties by the region with χ^2 smaller than $\chi^2_{min} + \chi^2_{torelance}$. We learn that constraints on PDFs are from constraining the upbound of the goodness of fits. When we fit theoretical models to match experimental data so that r_i for data points i and χ^2 are not too large, PDFs are constrained. Thus, we use χ^2 and r_i to see the constraints on PDFs because they represent the criteria of the goodness of fits and $f_a(\xi, \mu)$ values are constrained to meet this

Figure 1: Theoretical predictions and an experimental data point measurement with the error bar. Red crosses and blue crosses are two sets of theoretical prediction uncertainties.



criteria. In other words, the criteria could determine the range of $f_a(\xi, \mu)$. For instance, Fig. 1 is the comparison of two different fluctuations of theoretical values. Even though mean values of red crosses and blue crosses are the same, we can find the fluctuation of red crosses is easier to be detected because it's affection on residual values is larger than the fluctuation of blue crosses'. Fig. 2 is the comparison of two different fluctuations of residuals depending on $f_a(\xi, \mu)$. Although both of red circles and blue circles are strongly correlated, red circles are more sensitive to $f_a(\xi, \mu)$ because the $f_a(\xi, \mu)$ fluctuation of red circles strongly affects values of residuals. Therefore, To understand the relationship between data sets and the constraints on PDFs imposed by these data sets, we should study whether χ^2 and r_i are sensitive to the variation of $f_a(\xi, \mu)$ values.

C. Systematic method for studying constraints on PDFs imposed by experimental data sets

framework

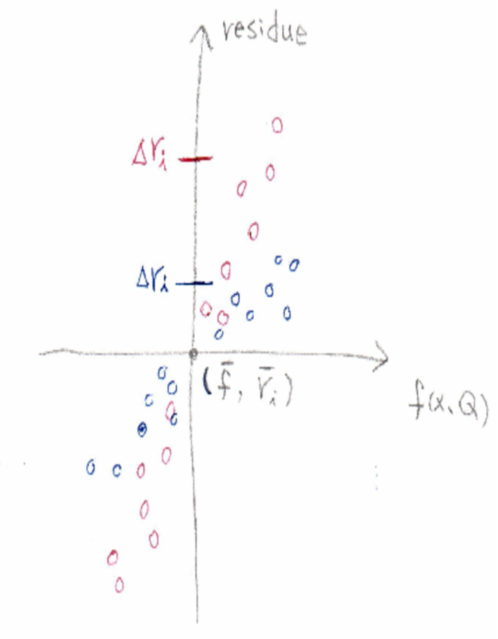
I have developed and tested a systematic method to study the constraints on PDFs imposed by experimental data sets. I use established statistical observables to quantify the strength of these constraints. After that, I introduce a statistical technique to visualize the regions of partonic momentum fractions ξ and QCD factorization scales μ where the experiments impose strong constraints on the PDFs. Recent experimental data is considered in the analysis in order to provide better constraints to various ranges of PDFs.

To test the effectiveness of the proposed method, I study constraints on CT14NNLO parton distributions [?] from various data sets. I include various types of experimental data sets in the analysis, including DIS processes, $Z \rightarrow l^+l^-, d\sigma/dy(l)$, $W \rightarrow l\nu$, and jet productions $(p_1p_2 \rightarrow jjX)$.

visualization method

For data sets of interest, we can demonstrate and identify values of correlation/sensitivity data by different colors on the $\xi - \mu$ plane $(2D - \xi - \mu$ figure), such as Figs. 2, which help

Figure 2: The sensitivity of a data point to a PDF. Red circles and blue circles are residuals of two data points versus f(x,Q) values in PDF error sets



us to rapidly estimate the distribution of the strength of constraints on the $\xi - \mu$ plane. We can also know the number of data points constraining PDFs by the histograms of the statistical quantities.

1. *Statistical quantities for constraints*

introduction correlation and sensitivity

Among various quantities that characterize the sensitivity of the experimental data to the PDFs, the correlations $Corr(f_a(\xi,\mu), r_i)$ of PDFs $f_a(\xi,\mu)$ and residuals r_i can determine whether there exist predictive relationships between PDFs and goodness of fit to data points. Here a is the flavour index, and r_i is the residual of data point i. We can also define a factor $\delta r_i \times Corr(f_a(\xi,\mu), r_i)$ to quantify the sensitivity of the experimental datum to a variation of the PDF. Both correlation and sensitivity are useful for constraining PDFs.

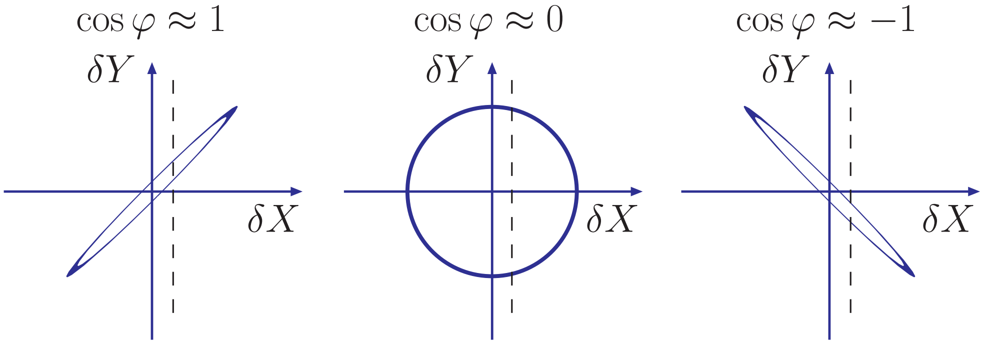
correlation

correlation's advantages

Correlation illustrates the strength of the predictive relation between any two observables X and Y. We can use values of one observable to predict values of another observable very well when their correlation is close to ± 1. correlations of Hessian uncertainties [?] have been used to see the simultaneous constraint on observables X and Y, and to get constraints on PDFs [?][?][?]. First, via measuring one physical observable, we are able to predict the value of another observable precisely. In addition, strong correlations are highly likely to show the signs of some physical relations, such as causation, between the two observables.

Hessian correlation: definition and physical meaning

Figure 3: Lissajous figure of observables X and Y for different $cos\phi$



$\cos\varphi \approx 1$

$\cos\varphi \approx 0$

$\cos\varphi \approx -1$

There are several ways to evaluate uncertainties on PDFs such as the Hessian method [?], the Monte Carlo method [?][?], and the Lagrange Multiplier [?]. Our PDF set input is CT14NNLO, which uses the Hessian method to estimate uncertainties information. This idea is based on the quadratic assumption. According to the quadratic assumption, we will get an elliptical shape of PDF parameter space around the best fit parameters $\vec{a_0}$ for a given tolerance parameter $\chi^2_{tolerance}$ satisfying $\chi^2(\vec{a}) < \chi^2(\vec{a_0}) + \chi^2_{torelance}$. If errors of an observable X along the ± directions of i-th dimension of the ellipse are X_i^+ and X_i^-, the uncertainty of X based on the variation of parameter at i-th dimension could be approximated by $(X_i^+ - X_i^-)/2$. According to the principle of error propagations, the X uncertainty via PDF parameter space is $\Delta X = \frac{1}{2}\sqrt{\sum\limits_i (X_i^+ - X_i^-)^2}$.

Our idea of studying PDF constraints from data sets uses the correlation between PDF Hessian sets and residual Hessian sets, where the Hessian correlation of two observables is defined as $cos\phi = \sum\limits_i (X_i^+ - X_i^-)(Y_i^+ - Y_i^-)/4\Delta X \Delta Y$. The correlation of any two observables X and Y could be used to see the simultaneous constraint of X and Y [?]. The ellipse of simultaneous constraint could be described by "Lissajous figure"

$$X = X_0 + \Delta X\, sin(\theta + \phi)$$

$$Y = Y_0 + \Delta Y\, sin(\theta + \phi)$$

where $0 < \theta < 2\pi$ traces the shape of the ellipse, and whether the shape is needle-like or circle-like is controled by ϕ. If $|cos\phi| \simeq 1$, the shape is needle-like, which strongly constrains Y for a given δX (see Fig. 3). Thus, the correlations $Corr(f_a(\xi,\mu), r_i)$ of PDFs $f_a(\xi,\mu)$ and the residuals r_i can determine the strength of constraints on PDFs imposed by r_i in experimental data points.

δr and sensitivity

construct more representative statistical quantity for constraints

Although we can know the predictivities between PDFs and measurements through $Corr(f_a(\xi,\mu), r_i)$, $Corr(f_a(\xi,\mu), r_i)$ could not specify the strength of constraints on PDFs imposed by r_i $(SOC(PDF))$. For instance, the measurements with large uncertainties cannot effectively constrain $f_a(\xi,\mu)$ no matter how large $Corr(f_a(\xi,\mu), r_i)$ is, since r_i is not sensitive to the variation of $f_a(\xi,\mu)$. Therefore, we want to find a more representative statistical quantity for $SOC(PDF)$. To study $SOC(PDF)$ between PDFs and data sets, we study the variation of χ^2 and r_i associated with the fluctuation in $f_a(\xi, \mu)$. Fig. 2 shows r_i in data points depending on the variation in $f_a(\xi, \mu)$ error sets. We find that despite the fact that the correlation between r_i in two data points (red circles and blue circles) and $f_a(\xi, \mu)$ are the same, the fluctuation for $f_a(\xi, \mu)$ imposes different levels of impact to r_i. The r_i, represented by red circles, are more sensitive to $f_a(\xi, \mu)$, which indicates that when we constrain χ^2 for getting the new fitted $f_a(\xi, \mu)$ error sets, the data point represented by the red circles will more dramatically narrow down the range of the new $f_a(\xi, \mu)$ error sets so that r_i for error sets become smaller. Here we find the δr_i, which evaluates the fraction of theoretical and experimental uncertainties, indicating whether the theoretical uncertainties are apt to be constrained after the fitting. For the above reasons, we advise using $\delta r_i \times Corr(f_a(\xi, \mu), r_i)$ to quantify the sensitivity $(Sen(f_a(\xi, \mu), r_i))$ for r_i to $f_a(\xi, \mu)$, and using the sensitivity to estimate $SOC(PDF)$ for data point i.

2. *Same point method*

objective

In principle, we can use the correlation & sensitivity mentioned above to quantify $SOC(PDF)$ for any points on the $\xi - \mu$ plane and data point i. Therefore, we can identify which regions in the $\xi - \mu$ plane have strong $SOC(PDF)$. Our objective is to characterize the strongly constrained ranges (Strong $SOC(PDF)$ Regions) imposed by the given data sets.

difficulties in the analysis and solutions

Acquiring the corresponding Strong $SOC(PDF)$ Regions for each point i still could not tell us which ranges are constrained by each data set because the amount of information in all data points is too large to analyze it. Therefore, we present a simple method as follows. For each data point i, we select the points (or the ranges) in the $\xi - \mu$ plane whose $f_a(\xi_i, \mu_i)$ are constrained most by the point i (Max $SOC(PDF)$ Regions). Here we assume that in scattering processes, sizes of physical observables are mainly contributed by the ranges near $\{\xi_i, \mu_i\}$. Thus, it is highly possible that the measurement at point i will impose the strongest constraints on the $f_a(\xi_i, \mu_i)$ in these ranges. As a result, the combination of these ranges describes the most constrained ranges for all data points in each data set.

capability

It is possible to evaluate those PDF ranges. For each experimental data point i, we can establish an approximate relation between the kinematical quantities for that data point, and unobserved quantities a, ξ, and μ specifying the PDFs, where a, ξ, and μ are flavour, momentum fraction, and resolution scale of partons. For example, in DIS, ξ and μ are approximately equal to Bjorken x and momentum transfer Q according to the Born-level kinematic relation. However, this relation is violated in high-order radiative contributions. Nevertheless, this relation will approximately hold in most scattering events. Therefore, we derive the relation between the kinematical quantities and unobserved quantities we mentioned for data sets in our analysis, including DIS, $dXsec/dy(l)$ of $Z \rightarrow l_+ l_-$, $dXsec/dy(l)$ of $W \rightarrow l\nu$, and (di)jet productions. In practice, our fitted PDF sets are not perfect, so even some ranges of the real PDF dominate a physical observable, the PDF sets in these ranges are not always strongly correlated to that physical observable.

Following are formulas (code part: selectExptxQv2 in II D 9) connecting experimental data points and their leading ξ, μ points of PDFs:

DIS: $x, Q_{data} = \xi, \mu_{PDF}$

$Z \to l_+l_-, dXsec/dy(l)$: $(Q/\sqrt{S}) \times \exp(\pm y), Q_{data} = \xi, \mu_{PDF}$

$W \to l\nu$: same as $Z \to l_+l_-, dXsec/dy(l)$

JP ($q_1q_2 \to j_1j_2$, estimate ξ_1, ξ_2 of jet as peak of $y(j_1), y(j_2)$): $(2P_T/\sqrt{S}) \times \exp(\pm y), P_{T_{data}} = \xi, \mu_{PDF}$

Here we give physics of these formulas. DIS processes are just mentioned in the example. In lepton pair production and (di)jet production, the rapidities of the final-state pairs are small for most events. If y is integrated out, we set $y = 0$ $\tau = Q/\sqrt{S}$ $\xi_1 = \xi_2 = \tau$. If y of the lepton pair or jet pair is known, we set $\xi_{1,2} = \tau \cdot \exp(\pm y)$. For jet production, $\tau = 2p_T^{\text{jet}}/\sqrt{S}$ at the leading order. In most events, if rapidity y_l of the lepton is known yet y of the boson is unknown, we use the fact that $y_l \sim y \pm 1$ for most events. You can still estimate ξ_1 and ξ_2 as $\xi_{1,2} = \tau \cdot \exp(\pm y)$, where $y \sim y_l$ (up to an error of less than 1 unit).

advantages

Because Max $SOC(PDF)$ Regions are obtained from a physical relation, we know the constraints in these ranges are the real physical constraints rather than the results of other factors, such as parametrization function dependency.

practical procedure (step by step)

Finally, we provide steps of Same point method. Steps are as follows:

1. calculate Max $SOC(PDF)$ Regions $\{\xi_i, \mu_i\}$ from corresponding experimental data points i by using a suitable transformation formula describing the approximate relation between $\{\xi_i, \mu_i\}$ and kinematical quantities for points i (code part: selectExptxQv2 in II D 9)

2. for each data point, calculate all flavours of PDF values for the same $\{\xi_i, \mu_i\}$ (executable part: fxQsamept.nb in III C)

3. calculate correlation $Corr(f_a(\xi_i, \mu_i), r_i)$, sensitivity $\delta r_i * Corr(f_a(\xi, \mu), r_i)$, and other statistical quantities ($r_{i,central\,value}$, δr_i, and experimental error ratio in this code) for every point i and every flavour a (executable: fxQsamept_corr.nb in III C)

4. draw the histograms and $2D - \xi - \mu$ figures for the statistical quantities derived in step 3 (executable: run_v3.nb in III C, example figure: 4)

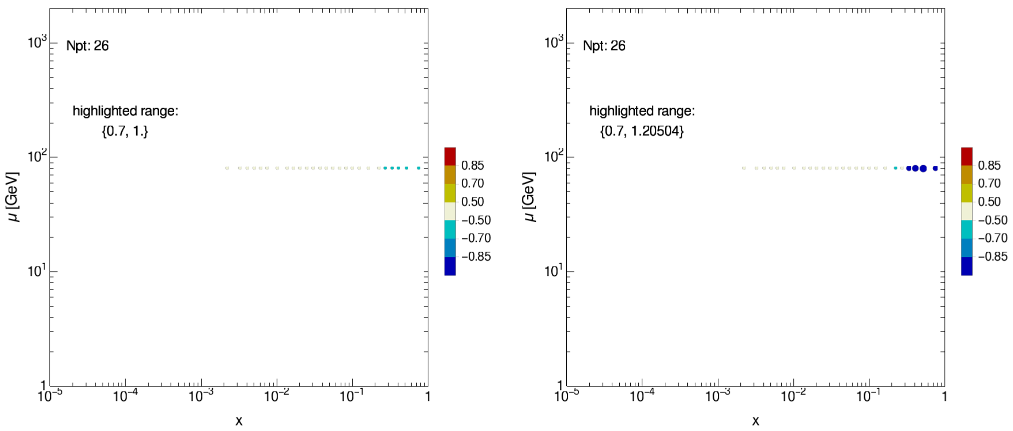
3. *Grid method*

eg version code: delete grid method part

Part II
Mathematica Program

II. STRUCTURE OF THE PROGRAM

The designing of the program will be introduced in this section. To know how the program works and how to operate it, we need to know its architecture II A, the convention of its data structure (format) II C, functions that users can use when they want to write their



((a)) The correlation $(Corr(d(x, \mu), r_i))$ of expt 281(D0 Run-2) and d quark.

((b)) The sensitivity $(\delta r_i * Corr(d(x, \mu), r_i))$ of expt 281 (D0 Run-2) and d quark.

Figure 4: The correlation and sensitivity of the Hessian uncertainty of data set 281(D0 Run-2) and CT14NNLO depending on the momentum fraction and the factorization scale.

../presentation/plots/proposal/CT14NNLO_225_corr_xQ_f2/samept.eps

../presentation/plots/proposal/CT14NNLO_225_corrdr_xQ_

((a)) The correlation $(Corr(d(x, \mu), r_i))$ of expt 225 (cdfLasy) and d quark.

((b)) The sensitivity $(\delta r_i * Corr(d(x, \mu), r_i))$ of expt 225 (cdfLasy) and d quark.

Figure 5: The correlation and sensitivity of the Hessian uncertainty of data set 225(cdfLasy) and CT14NNLO depending on the momentum fraction and the factorization scale.

../presentation/plots/proposal/CT14NNLO_247_corr_xQ_f0/gridpoints

../presentation/plots/proposal/CT14NNLO_247_corrdr_xQ_

Figure 6: The correlation and sensitivity of the Hessian uncertainty of data set 247(ATL7Zpt) and CT14NNLO depending on the momentum fraction and the factorization scale.

own executables II D, how to run existing executables, and what we expect to get from the existing executables III.

A. Leyers Of The Program

This program has three levels; In the layer 1 and 2 are functions which are called by the layer 3, and the layer 3 is the user interface. In the Fig.7 is a diagram of the architecture of the program.

Libraries: dtareadbotingw2016.m, pdfParsePDS2013

Layer 1: dtareadbotingw2016.m, pdfParsePDS2013

dtareadbotingw2016.m includes functions of 1. reading data from the .dta files. 2. reading information from the ExptID

pdfParsePDS2013 includes functions of inteporating PDF from data of .pds files

Layer 2: corr_proj_funcs.m

corr_proj_funcs.m defines the data structure II D 2, read and deals with data from .dta files II D 3, read and deals with PDF from .pds files II D 4, calculates observables (mainly correlation function) by data of .dta files and PDF II D 5, plots figures by data II D 7

Layer 3: This layer includes following kinds of programs

1. generate PDF data, residual data, and the .dta file of central set of selected $\{\xi, \mu\}$ values and flavours and save them into files

2. read data from files and calculate observables by data and use these observables to plot figures

3. tutorial of important functions in the program

4. show information of data in a data file or a PDF set (ex: included Experiments in this data or PDF set) and combine the selected two data

B. Processes Of The Program

C. Data Format

The program inputs data to draw figures on the $\xi - \mu$ plane. The program includes two kinds of executables. The first kind generate files of data; The second kind read data from files, and then calculate observables from data and draw figures of observables. data files contain info of the PDF values of all replicas, the residual values of all replicas, and the .dta file of central set. All observables in the program are listed in the following paragraph:

description of observables

fxQ: all family members of PDF $f(\xi, \mu)$

residual: central value of residual, formula of residual = (Theory - (shifted expt))/(unCorr error) (in every exptID of .dta file: ([[5]]-[[11]])/[[12]])

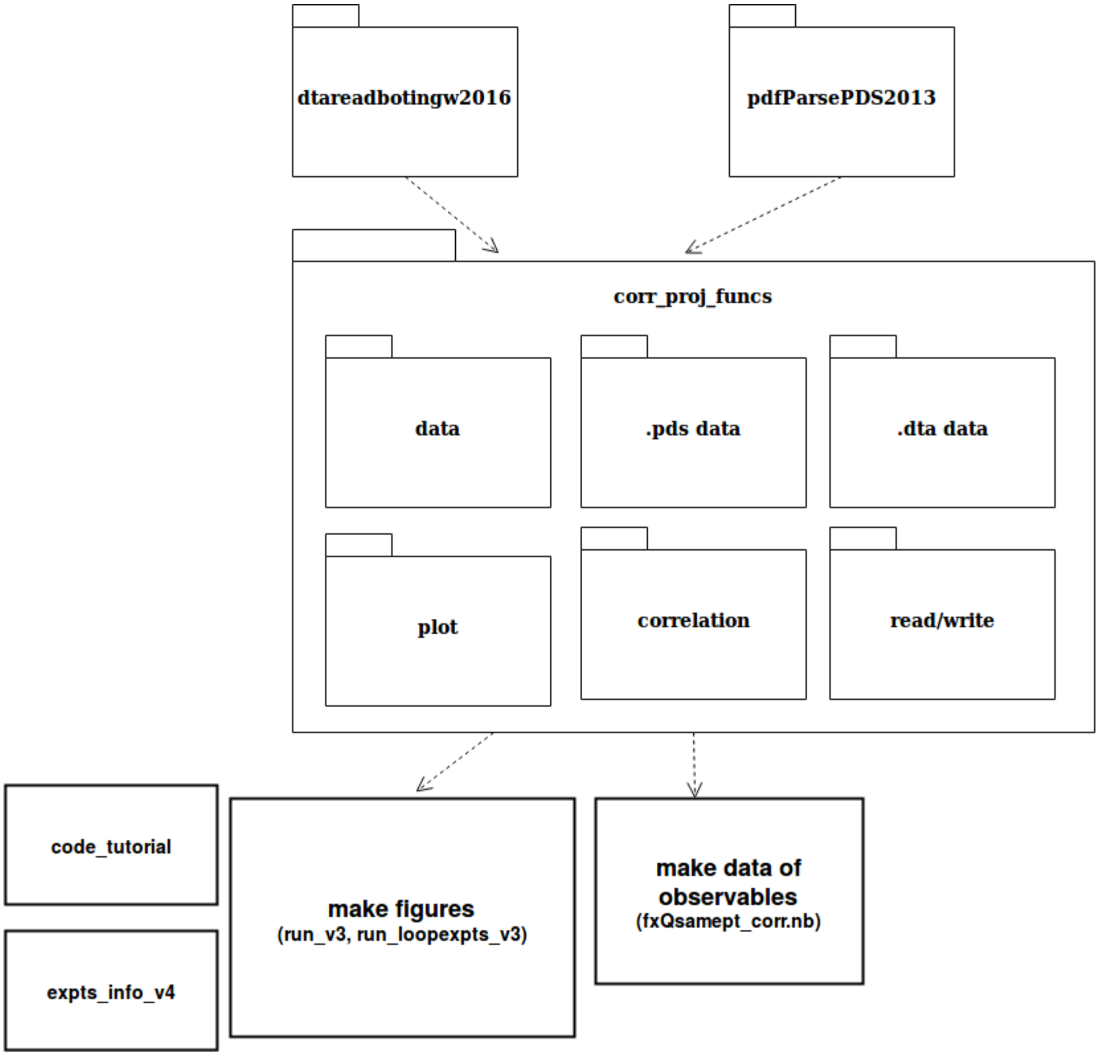
deltaR: Hessian symmetric uncertainty of residual

corr: correlation of PDF and residual

corrdr: deltaR*corr of PDF and residual

expt error/central: (expt error)/(expt central) (in every exptID of .dta file: [[6]]/[[4]])

Figure 7: the architecture of this program



TH error/central: (THeory error)/(Theory central), Theory error = Hessian asymmetric uncertainty of Theory

---

variable names of observables in Mathematica program

following are variable names representing these observables in the program. In different code files, observables may use different variable names.

fxQ: fxQsamept2class, fxQgridclass, fxQgrid2class

residual: residualdatacorrmaxclass, residualclass

deltaR: dRdatacorrmaxclass, deltaRclass

corr: corrdataclass, corrfxQdtaobsclass

corrdr: dRcorrdataclass, dRcorrfxQdtaobsclass

expt error/central: experrordatacorrmaxclass, expterrorclass

---

formats of data

The basic unit of a data is an Association with data and information of a data. Information contains experimental information (exptid, exptname, feyndiagram) and/or PDF information (PDFname, PDFsetmethod, Nset(total number of family members in this PDF set), iset (family member index of this PDF set), flavour). Data part contains data and data label. For example, fxQ[[iexpt,iflavour]] is a data of PDF of an experiment and a flavour, which looks like:

fxQ[[iexpt,iflavour]] =
<|
"label" -> {”x”,”Q”,”set1”,...”setN”},
"data" -> {LF[$\xi, \mu, f_1(\xi, \mu), ..., f_N(\xi, \mu)$],LF[$\xi, \mu, f_1(\xi, \mu), ..., f_N(\xi, \mu)$],......},
"exptinfo" -> Exptinfo,
"PDFinfo" -> PDFinfo
|>
Exptinfo=
<|
"exptid" -> 159,
"exptname" -> “HERA1X0”,
"feyndiagram" -> "unset"
|>
PDFinfo=
<|
"PDFname" -> "CT14NNLO",
"PDFsetmethod" -> "Hessian",
"Nset" -> 57,
"iset" -> "unset",
"flavour" -> "unset"
|>

The format of Data which are dealed with by functions of this program are always the basic unit or a List of it. For example, corrfxQresidualsamept[[“corrsamept”]][dtadataclassin_, fxQsameptdataclassin_] II D 5 inputs data of an experiment and data of a PDF, and then output a correlation data. The format of these inputs and output are just like the format of the example.

Format of data files for IO

When the program read or writes a data from/into files, it usually read/write data of many experiments and many flavours. In practical, the program read/writes a multi-dimensional List of the basic unit from/into one file. Every element of a List represents data of an experiment and/or a PDF flavour.

In this program, two methods I C 2 I C 3 are used to get data of observables. Which method is used to get a data determines the format of the data file. Following are formats of data files of samept observables and grid observables:

samept
f(x,Q) of all replicas:
list of [[iexpt,iflavour]]
data format=={LF[LF[$\xi, \mu, f(\xi, \mu)_0$,...,$f(\xi, \mu)_N$],LF[$\xi, \mu, f(\xi, \mu)_0$,...,$f(\xi, \mu)_N$],...}
residual of all replicas:
list of [[iexpt]]
data format=={LF[$\xi, \mu, r_{i,0}, ..., r_{i,N}$],LF[$\xi, \mu, r_{i,0}, ..., r_{i,N}$],...}
data in the .dta file of central set:

data format=={LF[val1, val2,...],LF[val1, val2,...],...} (formats are the same as in .dta files)

grid

eg version code: delete grid method part

D. Libraries and Functions

1. *global variables*

The global variables in this program are category of processes:

the rules of variable names is $var1$var2$var3...

$var1 = P, N, h (means proton, neutron or a complex nucleus is one of target in the collision);

$var2 = DIS, VBP, JP (deep inelestic scattering, vector boson production, jet production);

if $var2 = DIS, then

$var3 = NC, CC (neutral-current, charged-current);

$var4 = l, l1, l2, lqqbar, lccbar, lbbbar (lepton, one lepton, two lepton, $lq\bar{q}$, $lc\bar{c}$, $lb\bar{b}$ in the final state);

if $var2 = VBP, then

$var3 = Z, W ($Z$,$W$ boson process);

if $var2 = JP, then

no $var3 and ......

{PDISNC, hDISNC, NDISCC, hDISCC, PDISNCCC, PVBPZ, hVBPZ, PVBPW, PJP} are category of the main types of processes. these variables include Expt IDs with corresponding processes in a List of Mathematica.

2. *data class*

this section builds the structure of data and methods handling the data. The data are wrapped by function "Association" in Mathematica. Datamethods is an "Association" including methods dealing with data.

- dataclass: "Association" including data and it's information (experimental information or PDF set information), data are stored in dataclass[["data"]], the structure is {LF[e1,e2,e3,...],LF[e1,e2,e3...],LF[e1,e2,e3,...],...}, eN means the N-th element of LF[]. When set up the data in dataclass, the number of elements in all LF[] should be the same.
- Datamethods[["getdatainfo"]][dataclass]
  - return the information of a data
- Datamethods[["getdata"]][dataclass]
  - return the data part in dataclass

- Datamethods[[“setdata”]][dataclass, data]
  - set the data part of dataclass as data
- Datamethods[[“getNpt”]][dataclass]
  - return the number of points in the data
- Datamethods[[“getNcolumn”]][dataclass]
  - return the number of elements stored in a data point
- Datamethods[[“getdatalabel”]][dataclass]
  - return the label corresponding to elements of a data point. For example, the label of a 5-element data could be {”x”,”Q”,”Xsec”,”errorup”,”errorlow”}
- Datamethods[[“add”]][dataclass, dataadd, labeladd]
  - add new elements and in every data point and add the corresponding labels and return the dataclass, dataadd = {LF[e1,e2,...,eN],LF[e1,e2,...,eN],...}, labeladd={”lable1”,”label2”,...”labelN”}. the data and label of returned dataclass will have these N more elements and N more labels in the tail of every data point.
- Datamethods[[“pick”]][dataclass, picklist]
  - pick the specific elements in the picklist in a dataclass and return that dataclass. example: Datamethods[[“pick”]][dataclass, {3,1}], data in dataclass is {LF[e11,...,e13,...],LF[e21,...,e23,...],...,LF[eN1,...,eN3,...]}, then the function return {LF[e13,e11],LF[e23,e21],...,LF[eN3,eN1]}
- Datamethods[[“take”]][dataclass, takelist]
  - immitate “Take” function on every data points of the dataclass and return it, takelist could be the argument used in function “Take”
- Datamethods[[“delete”]][dataclass, deletelist]
  - Apply “Delete” function on every data points of the dataclass and return it, deletelist could be the argument used in function “Delete”
- Datamethods[[“tolist”]][dataclass]
  - return the data in dataclass as a List (LF -> List), ex: original data in dataclass is {LF[1,2,3],LF[10,20,30]}, then the function return {{1,2,3},{10,20,30}}
- Datamethods[[“picktolist”]][dataclass, picklist]

  - example: Datamethods[[“picktolist”]][dataclass, {3,1}], data in dataclass is {LF[e11,...,e13,...],LF[e21,...,e23,...],...,LF[eN1,...,eN3,...]}, then the function return {{e13,e23,e33,...eN3},{e11,e21,e31,...eN1}}

- Datamethods[[“LFglobal”]][dataclass]

  - Since the data generated by functions in dtaread2016boting.m file take the data structure dtaread2016boting‘Private‘LF[...], the structure should be changed to LF[...] so that some Mathematica skills can be used (such as replace /.)

3. .dta data class

this section is for dealing with .dta data

- Readdtafile[[“readdta”]][dtaDirin_, explistin_]

  - input the Directory of .dta files and a List of experimental ID, return the experimental data and information for all family members of the PDF set, the dimension of output: [[iexpt, iset]], the structure of returned object: if output of function “ReadExptTable” is {data1, data2,...}, the output[[iexpt, iset]] = dataN read from the iset-th .dta file, For example: Readdtafile[[“readdta”]][CT14NNLOpath, {101,201}], then the function returns output[[2,57]] for data of exptid = 101, 201

- Readdtafile[[“toclass”]][datain_, PDFnamein_, PDFsetmethodin_]

  - input a data (which is one experiment of output of function “ReadExptTable”), PDFname and PDFset method (present there is only “Hessian”), return the dataclass corresponding to this data. For example: Readdtafile[[“toclass”]][datain_, “CT14NNLO”,”Hessian”]

4. .pds data class

this section is for generating dataclass of PDF ($f(x, \mu, flavour)$) by calling pdfParsePDS2013.m in ./lib (see library: dtareadbotingw2016 for detail of this file).

- fxQcalculate[xQLFin_, PdsDirin_, PDFsetmethodin_, flavourin_]

  - input PDFDir, xQLF({LF[x_1, μ_1],LF[x_2, μ_2],...}), PDFsetmethod, and flavour (flavour = -5~5), return FxQdata class, with data structure {LF[$x_1, \mu_1, f(x_1, \mu_1, flavour, iset = 0), ..., f(x_1, \mu_1, flavour, iset = Nset)$],LF[$x_2, \mu_2, f(x_2, \mu_2, flavour, iset = 0), ..., f(x_2, \mu_2, flavour, iset = Nset)$],...}

- fxQsameptcalculate[dtadataclassin_, PdsDirin_, PDFsetmethodin_, flavourin_]

  - similar to fxQcalculate, the only difference is the source of points {LF[x_1, μ_1],LF[x_2, μ_2],...} are the data of dtadataclass (and return the FxQsameptdata), p.s. "exptinfo" is copy from Dtadata[["exptinfo"]], "PDFinfo" is set by input arguments

- setextrafxQ[fxQdataclasslistin_]
  - this function is specific for adding some system-defined combination of PDFs ($\bar{d}/\bar{u}$, d/u, $(s+\bar{s})/(\bar{u}+\bar{d})$). input an array of fxQdataclass which includes flavour from - 5 ~ 5, output the array with customized f(x, Q) (flavour = -5~8), flavour of 6, 7, 8 are defined as following : $\bar{d}/\bar{u}$, d/u, $(s+\bar{s})/(\bar{u}+\bar{d})$

5. *correlation data class*

this section is about calculating the correlation of residual and any function

- corrfxQresidualsamept[["corrsamept"]][dtadataclassin_, fxQsameptdataclassin_]
  - calculating $Corr(residual, f(x, \mu))$: input a dtadata class with data == {LF[ξ, μ, obs1, obs2, ... obsNset], ...} and a fxQsameptdata class with data == {LF[ξ, μ, f (ξ, μ, flavour, iset1), ..., f (ξ, μ, flavour, Nset)], ...}, the function assume $\{\xi, \mu\}$ of dtadataclass & fxQsameptdataclass are the same. output the corrsameptdata class with data structure {LF[ξ, μ, $Corr(obs, f(x, \mu, q))$], ...}
- corrfxQresidualsamept[["dRcorrsamept"]][dtadataclassin_, fxQsameptdataclassin_]
  - calculating $\delta r * Corr(residual, f(x, \mu, flavour))$: input a dtadata class with data == {LF[ξ, μ, obs1, obs2, ... obsNset], ...} and a fxQsameptdata class with data == {LF[ξ, μ, f (ξ, μ, flavour, iset1), ..., f (ξ, μ, flavour, Nset)], ...}, the function assume $\{\xi, \mu\}$ of dtadataclass & fxQsameptdataclass are the same. output the corrsameptdata class with data structure {LF[ξ, μ, $\delta obs * Corr(obs, f(x, \mu, flavour))$], ...}
- corrfxQresidualsamept[["deltaR"]][dtadataclassin_]
  - calculating δr: input a class with LF[ξ, μ, obs1, ..., obsNset], output δr as List : {$\delta r(1)$, $\delta r(2)$, ... δr (Npt)}
- corrfxQresidualsamept[["residual"]][dtadataclassin_]
  - calculating δr: input a class with LF[ξ, μ, obs1, ..., obsNset], output the same dataclass with data structure {LF[ξ, μ, δr], ...}

6. *read/write class*

This section is about read/write functions in the program.

- readcorrconfigfile4[configDirin_, configfilenamein_]
  - read arguments from a configure file. filename = "configDir/configfilename"
  - the configure file is for making figures executables
  - the configure file includes

- readsavedataconfigfile[configDirin_, configfilenamein_]
  - read arguments from a configure file. filename = “configDir/configfilename”
  - the configure file is for making data executables
- readplotdataconfigfile[configDirin_, configfilenamein_]
  - read arguments from a configure file. filename = “configDir/configfilename”
  - the configure file is for making figures executables
  - the configure file includes data filenames and data information of figures

7. *plot class*

this section is for figures output on the webpage; there are three kinds figure and one experiment name table

- PDFloglogplot[datain_, plotmarkerin_, plotstylein_, titlein_, xtitlein_, ytitlein_, plotrangein_, lgdin_, lgdposin_, imgsizein_]
  - plot the PDF data at $\xi - \mu$ plane with selected colors and markers to represent different experiments, the plot is log-log scale. data structure: {expt1,expt2,...}, exptN={LF1[ξ, μ],LF1[ξ, μ]}
  - detail of argument setting is in the position of function definition in the program
- PDFCorrelationplot8[datain_,titlein_,xtitlein_,ytitlein_,plotrangein_,stretchxin_,stretchyin_,bar
  - make plot of the data on $\xi - \mu$ plane, color of the point of the data depend on value of data at that point, data size, highlighted data will become larger data here is a List of data: {data1, data2, ...}. data structure of dataN = {LF1[ξ, μ,value],LF1[ξ, μ,value],...}
  - points of data1, data 2, ... will have different shape
  - detail of arguments setting is in the program
- histplot4[histlistin_, titlein_, xtitlein_, ytitlein_, binsetin_, lineelementin_, plotrangexin_, Nbinin_]
  - make histogram of data value: input the List of value and other arguments, return a histogram. data structure = {value, value,...}
  - detail of arguments setting is in the program
- makeGrid2[strin_, rowsin_, titlein_]
  - make experiments name table: input List of exptname, #rows per column, title of table, return a table of expt name
  - ex: makeGrid2[{”name1”,”name2”,”name3”,”name4”,”name5”}, 3, “title example”]

the function plotting all kinds of figures for the webpage output

- processdataplotsmultiexp6percentage[corrfxQdtaobsclassin_, configargumentsin_, plottypein_, flavourin_]
  - make plots of the selected figure type: the function will return one $\xi - \mu$ plot and two histogram for figure type (plottype) = 2~6 and one $\xi - \mu$ plot for figure type = 1
  - data is set by corrfxQdtaobsclass, for figure type 2, 3, 4, data structure is: dataclass[[iexpt]]; figure type = 5,6, data structure = dataclass[[iexpt,iflavour]]; for figure type = 1, data structure is the same as figure type 5, 6, the function will extract the (ξ, μ) from the data
  - figure type option(the same as in config file): # 1: data plots, 2: expt error, 3: residual, 4: "residual error" deltaR_i, 5: "sensitivity factor" deltaR_i*Corr(r_i, F), 6: "correlation" Corr(r_i, F)
  - configarguments is the output of configure file, it is convenient to input arguments in config file for making plots
  - when the figure type = 5, 6, the flavour index should be set (convention of the flavour index is in ".pds data class" section, and the index of user defined function is 9)
  - example: flavour =0; p = processdataplotsmultiexp5percentage[dRcorrfxQdtaobsclassfinal, readcorrconfigfile4[configDir, configfilename], 5, flavour];

8. other class

This section is for all functions that are hard to classified

- implementeps[PlotDirin_, DirTypein_]
  - convert eps files into one pdf file
  - eps files are output of making figure executables
  - input : Diretory storing figures, DirType == "samept", "grid", "sameptgrid", users should setup the right type for the method he uses to produce figures
- implement[PlotDirin_, DirTypein_]

9. library: dtareadbotingw2016

this library is about 1. reading data from the .dta files. 2. reading information from the ExptID
(*perhaps need to clean unused functions*)

- ReadLisFile[lisFileName]

  - input the file of CTEQ expt list format (ex: filename = dat16lisxxx), return a lisTable, which record the information of experiments. This lisTable should be activated before some functions in dtareadbotingw2016 are used

- ExptIDtoName[ExptIDin_]
  - return the name of ExptID
- ExptIDEcm[ExptIDin_]
  - return the E_{cm} of the ExptID
- ExptIDprocess[ExptIDin_]
  - return the information of the ExptID (presently the information includes Expt name, Feynman diagram, E_{cm})
- ExptIDinfo[ExptIDin_]
  - return the type of the process (“DIS”, “VBP1”,“VBP2”,“VBP3”,"JP" are used in present calculation. other types can be found in function)
- ReadDta[DtaFileName]
- ReadExptTable[DtaFileName, FileFormat]
  - this function input the .dta file and the file format ("ct2016", "ct66", "ct60"). For CT14NNLO .dta, the file format is "ct2016"
- selectExptxQv2[ExptIDin_, datain_, Sin_] (*Sin is not used, should be modified*)
  - input the ExptID and data (data structure = {LF[a__],...}), then calculate the most possible (ξ, μ) of the PDF depend on Expt type (output of ExptIDinfo). return the data with the structure = {LF[a__, ξ, μ],...}. formulas are explained in I C 2.

type of processes:

DIS: deep inelastic scattering

VBP1: Q, τ (???)

VBP2: $Z \to l_+ l_-, dXsec/dy(l)$

VBP3: $W \to l\nu$

JP: jet production, $q_1 q_2 \to j_1 j_2$ (estimate x_1, x_2 of jet as peak of $y(j_1), y(j_2)$)

Expt ID in types:

DIS = 100~199;

VBPtype1 = {};

VBPtype2 = {201, 203, 204, 260, 261, 268, 240};

VBPtype3 = {225, 227, 234, 267, 281, 241, 266};

JP = 500~599

formulas:

DIS: $x, Q_{data} = \xi, \mu_{PDF}$

VBP1: $Q/\sqrt{S}, Q_{data} = \xi, \mu_{PDF}$

VBP2: $(Q/\sqrt{S}) \times \exp(\pm y), Q_{data} = \xi, \mu_{PDF}$

VBP3: same as VBP2

JP: $(2P_T/\sqrt{S}) \times \exp(\pm y), P_{T\,data} = \xi, \mu_{PDF}$

derivation of formulas:

(* need modify) By center-of-mass relation: $\hat{s} = \xi_1\xi_2 s$, $\xi_1 \times \xi_2$ could be decided. Assuming the ξ_1, ξ_2 values are close to the peak of some functions such as $y(jet)$, the ξ_1, ξ_2 value could be decided.

III. HOW TO RUN THE PROGRAM

A. Run Step By Step

Executables can be classified into making figures and others. Making figures is the main part of this program. The input/output and processes of Making figures executables will be introduced in this section. Other executables are for tutorial of main functions (code_tutorial.nb) and showing information of data (expts_info_v4.nb). expts_info_v4.nb makes users accessible to the information of data used to draw figures and the analyzed directory with .dta files. This section will focus on making figures executables.

Steps of running the program

1. edit "savedata_config.txt", then run making PDF data exoecutables (see III C)
2. edit "savedata_config.txt", then run making observable data executables
3. edit "config1.txt" (and "plotdata_config.txt"), then run making figures by observable data executables

Let's take Samept method as example

1. edit "savedata_config.txt", then run fxQsamept.nb
2. edit "savedata_config.txt", then run fxQsamept_corr.nb
3. edit "config1.txt" (and "plotdata_config.txt"), then run run_v3.nb

Output figures are in ./plots directory

B. Processes Of Executables

There are three kinds of executables: making PDF data, making observable data, and making some figures of data which are produced by the program. The first and the second kinds of executables read arguments from “savedata_config.txt”; The third kind of executables read arguments from “config1.txt”.

To conveniently explain processes of these executables, using flowcharts to visualize processes is a simple way. Fig.8, Fig.9, and Fig.10 are flowcharts of three kinds of executables:

C. Input And Output Of Executables

There are three kinds of executables: making PDF data, making observable data, and making some figures of data which are produced by the program. The first and the second kinds of executables read arguments from “savedata_config.txt”; The third kind of executables read arguments from “config1.txt” and “plotdata_config.txt”. Following are descriptions of their executables, input, and output (as Table I shows)

To make data, users need to set up arguments in savedata_config.txt:

- Making PDF data
  - executables: fxQsamept.nb
  - arguments that users should set up
    - PDF set Dir, PDF method, Expt ID List
    - F(x,Q) Samept Path & F(x,Q) Samept File for samept f(ξ, μ) data
    - F(x,Q) Grid Path & F(x,Q) Grid File, Nx, NQ for grid f(ξ, μ) data
    - F(x,Q) Grid Path & F(x,Q) Grid File, Nx for xgrid f(ξ, μ) data (Q values are temporary set as Z mass and W mass, this part should be modified)
  - output: if Path & File are “default”, the program make a file:
  - ./quick_data/fxQ_{$methods}_{$PDFname}.dat for samept data executables, ./quick_data/fxQ_{$methods}_{$PDFname}_x{$Nx}_{$NQ}.dat for grid data executables, ./quick_data/fxQ_{$methods}_{$PDFname}_x{$Nx}.dat for xgrid data executables, where $PDFname = PDFname, where $methods = “samept”, “grid”, “xgrid”, $PDFname = PDFname, ex: CT14NNLO

- Making observable data
  - executables: fxQsamept_corr.nb
  - arguments that users should set up
    - PDF set Dir, PDF method, Expt ID List, datalis file
    - Correlation Path & Correlation File
    - F(x,Q) Samept Path & F(x,Q) Samept File for samept f(ξ, μ) data
    - F(x,Q) Grid Path & F(x,Q) Grid File, Nx, NQ for grid f(ξ, μ) data
  - output: if Path & File are “default”, the program make files with all $obs indexes:

Figure 8: the flowchart of making PDF data

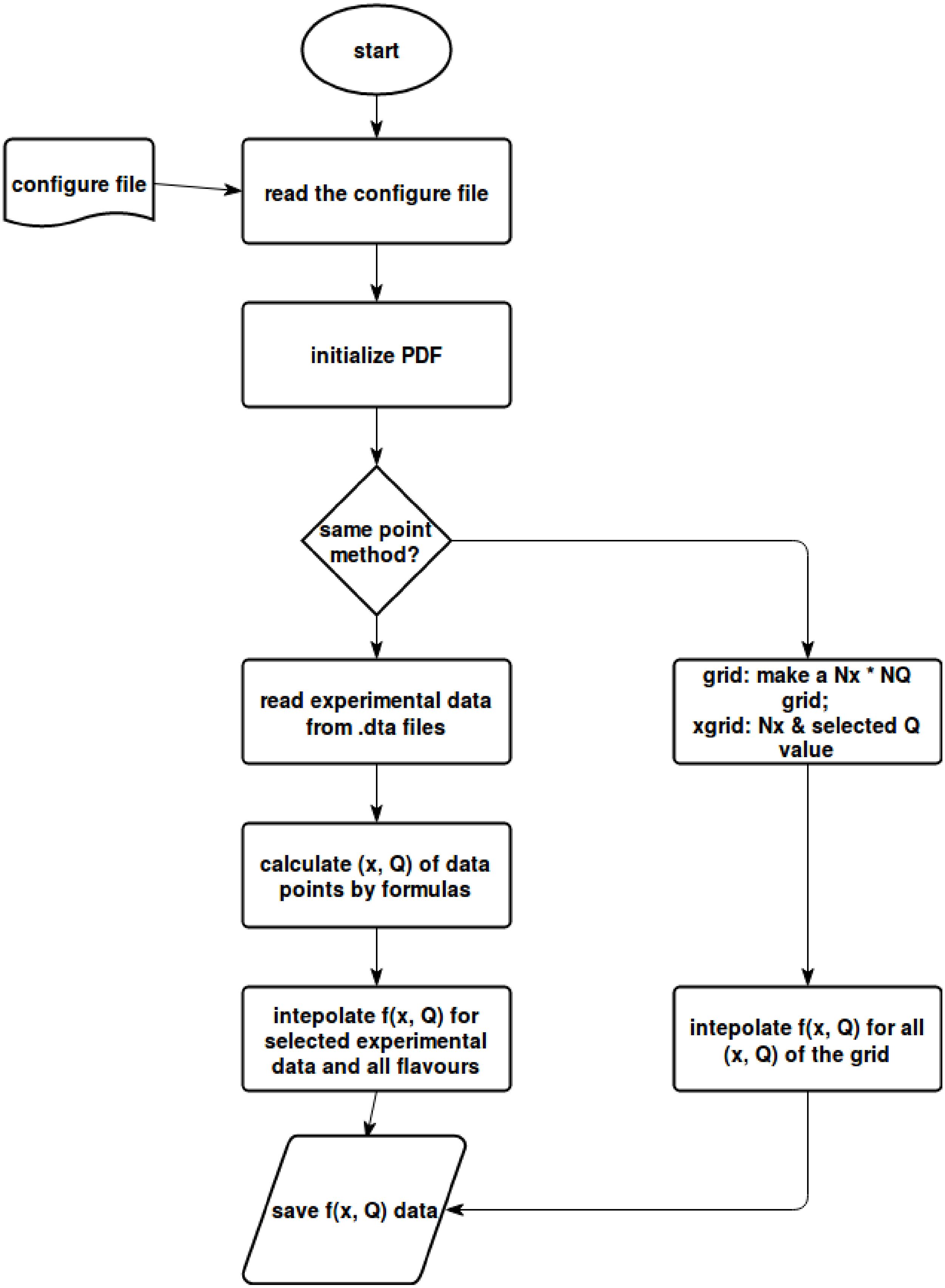

start
configure file
read the configure file
initialize PDF
same point method?
read experimental data from .dta files
grid: make a Nx * NQ grid;
xgrid: Nx & selected Q value
calculate (x, Q) of data points by formulas
intepolate f(x, Q) for selected experimental data and all flavours
intepolate f(x, Q) for all (x, Q) of the grid
save f(x, Q) data

Figure 9: the flowchart of making observable data

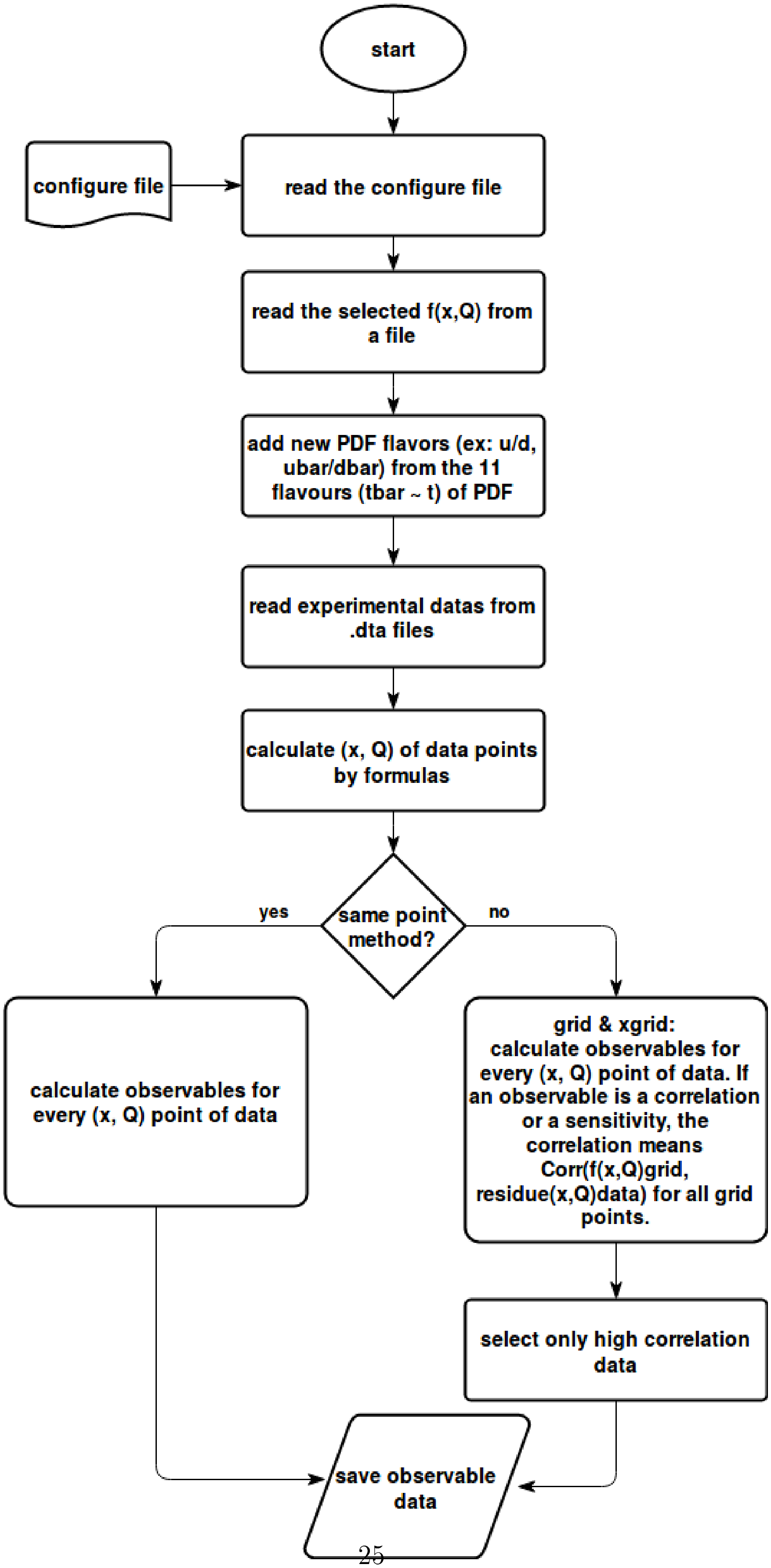

start
configure file
read the configure file
read the selected f(x,Q) from a file
add new PDF flavors (ex: u/d, ubar/dbar) from the 11 flavours (tbar ~ t) of PDF
read experimental datas from .dta files
calculate (x, Q) of data points by formulas
same point method?
yes
no
calculate observables for every (x, Q) point of data
grid & xgrid: calculate observables for every (x, Q) point of data. If an observable is a correlation or a sensitivity, the correlation means Corr(f(x,Q)grid, residue(x,Q)data) for all grid points.
select only high correlation data
save observable data

Figure 10: the flowchart of making figures

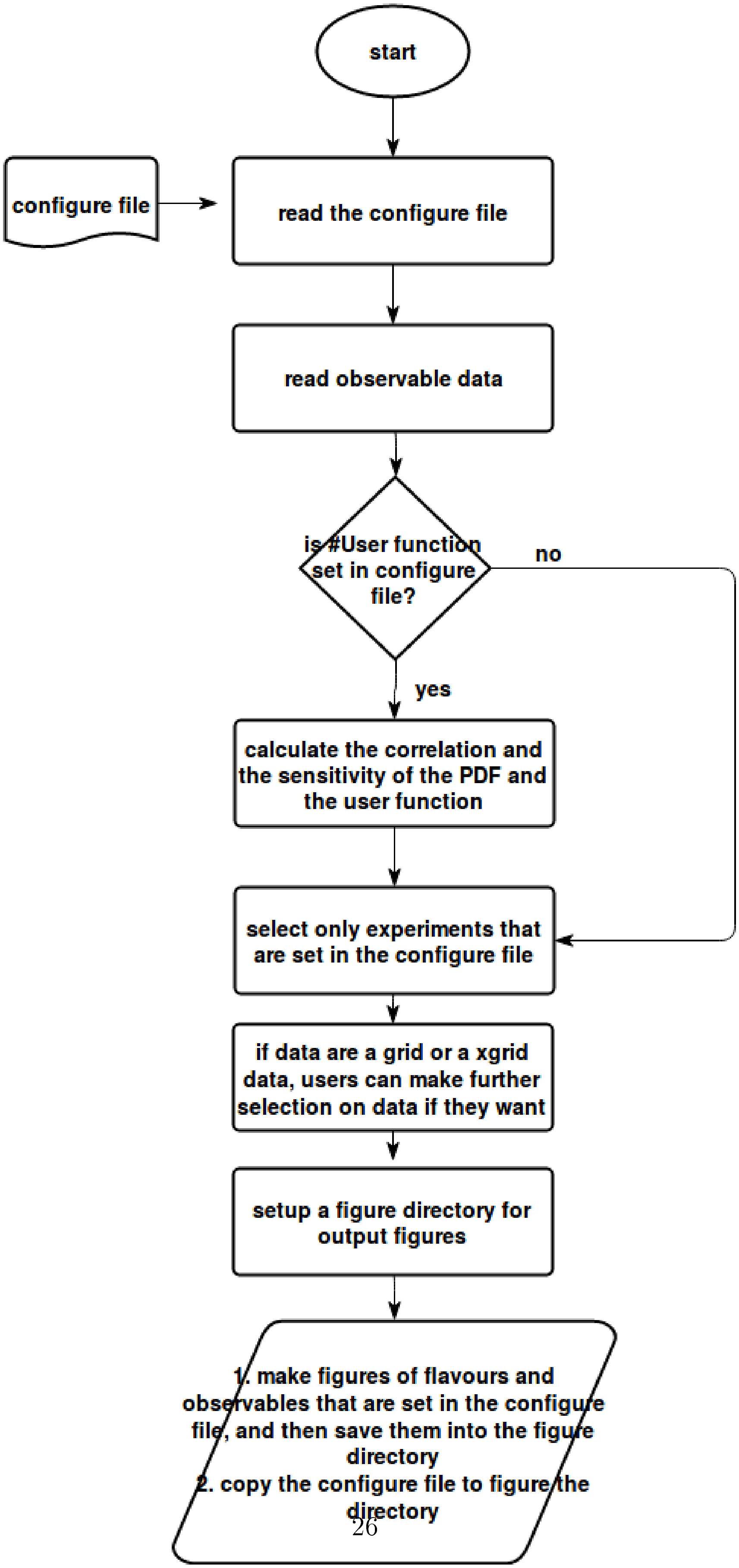

start
configure file
read the configure file
read observable data
is #User function set in configure file?
no
yes
calculate the correlation and the sensitivity of the PDF and the user function
select only experiments that are set in the configure file
if data are a grid or a xgrid data, users can make further selection on data if they want
setup a figure directory for output figures
1. make figures of flavours and observables that are set in the configure file, and then save them into the figure directory
2. copy the configure file to figure the directory

Table I: functions of executable files by table: columns are methods of making plots; rows are steps of methods.

	Making observable data, configure files: savedata_config.txt	Making figures, configure files: config1.txt and plotdata_config.txt
Samept method	fxQsamept_corr.nb (.m)	run_v3.nb (.m)

  - ./quick_data/{$obs}_samept_data_{$PDFname}.dat for samept data executables, where $PDFname = PDFname, ex: CT14NNLO, $obs = "corr", "dRcorr", "dR", "residual", "residualNset", "expterror"

To make figures, users need to set up arguments in config1.txt & plotdata_config.txt:

- Making figures by observable data
  - executables: run_v3,
  - arguments that users should set up
    - #PDF set, # Figures to plot, # Experiments to include, #Functions to use in correlations, #User function parameters are for setup of the input data
    - #x-Q figure parameters, #Histogram figure parameters, #in plots, #highlight mode, #data point size are for setup of how figures look like
    - for grid & xgrid executables, Path & File of data, Nx, NQ are temporary set in the code (this part should be modified)
  - output: 2D-xQ, histograms

D. SOP of Plotting Figures of New Experiments

When a user want to plot figures of new experiments, he should check whether they are given some (x,Q) transform formulas. The user need to check ExptIDEcm & ExptIDinfo in dtareadbotingw2016.nb, which determine which formula should be applied on an experiment and the center-of-momentum energy $\sqrt{S}$ of that experiment. We can find from II D 9 that some formulas contain information of center-of-momentum energy, which should be set in ExptIDEcm.

Following are the steps of plotting figures of new experiments:

1. set up the $\{\xi, \mu\}$ transform formulas and $\sqrt{S}$ of new experiments (selectExptxQv2 and ExptIDEcm in II D 9)

2. edit configure files and run fxQsamept.nb to get data of PDF for $\{\xi, \mu\}$ of these experiments

3. edit configure files and run fxQsamept_corr.nb to get data of observables for $\{\xi, \mu\}$ of these experiments

4. edit configure files and run run_v3.nb to get figures of observables of these experiments

Appendix A: update notes

version 22:
make script versions for executables:

- script executables that have been made: all executables of making figures
- run them in terminal: “math -script xxx.m” under bin directory

delete unused functions:

- add functions: implementeps[PlotDirin_, DirTypein_], implement[PlotDirin_, DirTypein_], readplotdataconfigfile[configDirin_, configfilenamein_] (configure file is plotdata_config.txt inIII C)

data files of executables that making figures are set up in "plotdata_config.txt"
version 23:

- fix the error text: residue -> residual
- delete resolution part in physics chapter I B 3

version 23_v7:

- add formulas of dividing the grid for Grid method I C 3
- add a table to describe executable files **??**
- add the description of grid method for $Q = M_W$, M_Z processes I C 3

version23_eg_v8:

- add descriptions of expt_info_v4.nb
- add new subsection of steps of running the program
- renew figure **??**
- saved data file from extensions from .m to .dat

version23_eg_v10:

- rewrite the same point method part

version23_eg_v11:

- delete descriptions about grid method

Appendix B: eg version

In mathscript_v23_eg (example version):

1. there are only Samept method executables in this version for simplification of codes.
2. fxQsamept.nb and fxQsamept_corr.nb are combined into fxQsamept_corr.nb; That means fxQsamept_corr.nb produces observable data directly from data in .dta files and PDF values in .pds files
3. to making figures: run fxQsamept_corr.nb and run_v3.nb
4. users can read code_tutorial.nb to understand important functions in this program
5. expts_info_v4.nb could be used to see information in observable data files and .dta files (such as expt IDs contained in the file)[?] [?]

References