

ManeParse: a set of Mathematica packages for reading and calculating with LHA and CTEQ Parton Distribution Functions

D. Benjamin Clark, Eric Godat, Fred Olness*

Department of Physics, Southern Methodist University, Dallas, TX 75275-0175, USA

Abstract

We describe ManeParse, a set of Mathematica packages for use with LHA and CTEQ Parton Distribution Functions (PDFs). ManeParse is made up of four packages. The first package, pdfCalc.m, is the core package. It houses the interpolation and manipulation functions for both CTEQ .pds files and LHA .info and .dat files. Interpolation is needed to smooth the provided PDF grids into continuous functions. These PDFs are needed for QCD calculations of hadronic interactions at colliders such as the LHC. The second package, pdfParseLHA.m, reads in LHA style .info and .dat files from numerous collaborations such as CTEQ, MSTW and NNPDF and provides the user access to nuanced functions designed to utilize LHA formatting agnostic of the collaboration as well as functions that enable the user to optimally use a specific collaboration's data. The third package, pdfParseCTEQ.m, reads in CTEQ .pds files and provides the user with functions that specifically target that file format and includes retrofitting for older generations of CTEQ formatting. The fourth package, pdfErrors.m, allows the user to calculate the theoretical uncertainty, parton-parton Luminosity, and Correlation for Hessian and Monte Carlo PDF sets. A demo file is provided along with a set of example PDF sets.

Keywords: QCD; Mathematica; Parton Distribution Functions; PDF; PDFs; Hadron collider; PDF Errors; Hadronic Cross Section

PROGRAM SUMMARY

Manuscript Title: ManeParse: a set of Mathematica packages for reading and calculating with LHA and CTEQ Parton Distribution Functions and their uncertainties.

Authors: D. Benjamin Clark, Eric Godat, Fred Olness

Program Title: ManeParse

Journal Reference:

Catalogue identifier:

Preprint number:

Licensing provisions: none

Programming language: Mathematica package

Computer: all

Operating system: all

RAM: <1 MB per file

Supplementary material:

Keywords: QCD; Mathematica; Parton Distribution Functions; PDF; PDFs; Hadron collider; PDF Errors; Hadronic Cross Section

Classification:

Nature of problem: CTEQ PDFs are currently read and interpolated via a FORTRAN interface. LHA PDFs are available for multiple collaborations but rely on a FORTRAN and C++ interface from LHAPDF software. This set of packages provides for the implementation and use of LHA and CTEQ PDFs for hadronic interaction calculations in the Mathematica environment.

Solution method: Using Mathematica's file read ability, a family of PDFs can be read and parsed, via the parsing package that matches the file type of interest, into a three dimensional array in Bjorken x , the hard scattering energy Q , and parton flavor in the LHA numbering scheme. The Calc package is then capable of interpolating the PDF central value. The Error package allows for calculation of the uncertainty on the PDF

*E-mail address: dbclark@smu.edu, egodat@smu.edu

or a function of the PDFs in a variety of ways. The Error package is also capable of calculating the correlation between functions of PDFs and parton-parton Luminosities

Running time: Depends on details of the calculation and the required numerical accuracy. During testing the pdfParse package was capable of reading and parsing 654 PDS files in 46 seconds, while the pdfParseLHAall package was capable of reading in a full LHA PDF family (1 .info file and 53 .dat files) in 2 seconds. Details of testing can be found in this paper.

1. Introduction

Currently, Parton Distribution Functions (PDFs) are read in and interpolated using Fortran or C++ interfaces. A Mathematica interface has been requested by researchers in the particle physics community. This paper describes a set of Mathematica packages capable of reading CTEQ .pds files and LHA .info and .dat files from numerous collaborations and interpolating to provide a PDF at any value of Bjorken x and hard scattering energy Q . In addition, the packages provide a simple interface for the calculation of theoretical uncertainties on the PDFs and functions of the PDFs for both Hessian and Monte Carlo PDF families.

The paper is organized as follows: Section II discusses the motivation for the package along with the main strategies adopted in its design. Section III describes a bug in the CTEQ 9 and beyond distributions and the fix provided. Section IV gives examples of simple calculations and output from the package.

An Appendix is provided detailing the functions available in the packages.

2. Strategy and Design

When writing this set of Mathematica packages, the main goals of this project were to create a package that is reliable, fast and transparent. ManeParse achieves these goals with a set of lightweight packages that are capable of very quickly loading a large number of PDFs into memory and using a custom four-point Lagrange interpolation routine. The interpolating routine is provided once in the code and called each time it is needed. Transparency of the code is achieved by avoiding the built-in Mathematica interpolation routines. These routines are a black-box and, although they are powerful in a variety of applications, they are far too slow for the present application. The interpolation routine is fast and provides immediate access to the PDF at any value of Bjorken x and hard scattering energy Q within the bounds of the PDF set. The user then has access to the powerful computational and plotting tools available in Mathematica to perform calculations of hadronic interactions.

As noted in the introduction, CTEQ PDFs are currently read and interpolated via a FORTRAN interface. This interface is difficult to read and contains many instances of repeated code. In addition, it is very difficult to link this code to Mathematica meaning there is no reliable way to interface the CTEQ .pds files with Mathematica for calculations. It is worth noting that the current FORTRAN CTEQ interface also uses a four-point Lagrange interpolation so that the output from the Mathematica packages for .pds files is the same as what would be obtained from the FORTRAN interface.

LHAPDF is also FORTRAN based but includes a C++ interface, and requires installing LHAPDF software in order to interact with the PDF files. However LHA-style .data and .info files are available from HepForge and the packages allow the user to directly interact with those files through Mathematica. Multiple collaborations have provided data sets in this format and the package is equipped to handle the subtle variations among them.

3. Bug Report

While developing this package, a bug was discovered in the CTEQ PDF data files. Double precision calculations were introduced with CTEQ 9. This allows for a three digit exponent in the data file for PDFs. When the .pds files were generated, all data with a two digit exponent is printed in the proper format with an “E” between the mantissa and exponent (e.g. 2.16855E-10). However, when the data

calls for a three digit exponent, the data is missing the “E” between the elements (e.g. 2.31434-108). This bug is produced by the FORTRAN compiler used for the generation of .pds files. The bug has not been detected previously as the compiler makes the error when generating the files and then corrects the error when reading them. If the user is reading the code with the provided FORTRAN interface, the bug is invisible to the user. However, when using the Mathematica interface the bug produces erroneous results.

The authors have provided a short Perl script (noe2.perl) that allows the user to fix these errors within the files in an automated way. The script will generate corrected files and produce an output log for the user informing her of the specific location of the bad data and the corrected output. Files can be run through this script in batch with a simple shell script quickly producing a set of error free files for reading and parsing by the Mathematica interface. Of note is that the repaired files remain able to be read in by the original FORTRAN interface.

Example: In the command line, input:

```
perl path_to_script/noe2.perl path_to_directory_containing_files/*
```

This would generate output of the form:

```
# 3211 =====
***** DIFFERENT
9.28334E-10 4.77089E-10 2.16855E-10 6.54559E-11 0.00000E+00 2.31434-108
-----
9.28334E-10 4.77089E-10 2.16855E-10 6.54559E-11 0.00000E+00 2.31434E-108
# 3212 =====
***** DIFFERENT
4.04261-129 2.59883-118 2.31434-108 3.32320E-99 8.88354E-91 5.04683E-83
-----
4.04261E-129 2.59883E-118 2.31434E-108 3.32320E-99 8.88354E-91 5.04683E-83
#5099 =====
All Done!
```

Here we see that there were four instances of the bug in this file and they have been repaired by the script.

4. Global Variables

The following are global variables set in the pdfCalc.m package and used by all packages.

- nSetCount
 - The number of PDF sets read into memory.
- pdfTableData
 - Nested list containing the parsed data from the PDF sets.
- alphalist
 - List containing the α_S data from the PDF sets.
- pdfxmin
 - List of minimum values of Bjorken x for the PDF sets.
- pdfSetList

- Prints the list of the loaded .pds/.dat grid files, their setNumber, maximal number of quark flavors, and number of valence flavors in each PDF set
- numQpart
 - Number of partitions in a given set. These partitions occur at the quark masses for some collaborations.
- alphaSFitParameter
 - A parameter used to generate α_S data from the PDF sets that only provide a single α_S value at a fixed energy, Q .
- infoTable
 - A list of rules and values that contains the information from the header or .info file that corresponds to each PDF set.

5. Appendix

The package pdfCalc.m provides the following functions:

- pdfAlphS[setNumber, Q]
 - This function returns the value of α_S at hard scattering energy Q when this information is available in the .pds or .info file.
 - If the file gives a single value for α_S at Q_{fixed} , the values will be interpolated using $\alpha_S = \beta/Log[Q_{fixed}]$ to solve for β . Then $\alpha_S(Q) = \beta/Log[Q]$
 - *Warning:* This function will print a text message and return a Null value if the α_S information is not available.
- pdfFlavor[flavor]
 - This function accepts an integer *flavor* number and returns a string with the name of the flavor in the LHA scheme.
 - Example: *pdfFlavor*[0], *pdfFlavor*[1], *pdfFlavor*[2] will return “gluon”, “down”, “up” for the gluon, down quark, and up quark PDFs.

flavor	parton
-6	tbar
-5	bbar
-4	cbar
-3	sbar
-2	ubar
-1	dbar
0 or 21	gluon
1	down
2	up
3	strange
4	charm
5	bottom
6	top

- The flavor can be specified with the following inputs:
- pdfFunciton[setNumber, flavor, x, Q]
 - This function returns the interpolated value of the PDF for the .pds/.dat file specified by *setNumber*, for the given flavor and value of Bjorken x and scale Q .
 - *Warning:* The results of this function are only reliable between the maximum and minimum values of x and Q in the .pds/.dat file.

- pdfGetInfo[setNumber]
 - This function returns the information corresponding to set *setNumber* read from the .info file or generated from the header of a .pds file.
 - pdfGetInfo[setNumber, value]: This function accepts a string and returns the info corresponding to set *setNumber* read from the .info file or generated from the header of a .pds file for a specific *value*.
 - Example: *pdfGetInfo[setNumber, “Flavors”]* will return the quark flavor scheme for the info file if that information is available.
 - *Note*: If the user is unaware of what is present in the info file, *pdfGetInfo[setNumber]* may still be used and displays the all values in the .info file.
- pdfGetQlist[setNumber]
 - This function returns the grid values in *Q* from the PDF set *setNumber*.
 - This function has the ability to see the grid values for data files with multiple grids.
 - For this use: *pdfGetQlist[setNumber, Qpartition]*
- pdfGetXlist[setNumber]
 - This function returns the grid values in *x* from the PDF set *setNumber*.
 - This function has the ability to see the grid values for data files with multiple grids.
 - For this use: *pdfGetXlist[setNumber, Qpartition]*
- pdfLowFunction[setNumber, flavor, x, Q, [power]]
 - This function returns the value of the PDF as in pdfFunction, but with an extrapolation below the minimum *x* value that goes as $1/x^{power}$.
 - The optional input, *power*, has a default value of *power* = 1.0.
- pdfNumQpartition[setNumber]
 - This function returns the number of *Q* grids in the PDF set *setNumber*.
- pdfResetCTEQ[]
 - This function deletes all .pds/.dat files from memory and resets all the internal variables in the package.
 - *Note*: This function does not accept any inputs.
- pdfSetListDisplay[]
 - This function provides the user a formatted table of the list of data sets stored in memory.
 - *Note*: This function does not accept any inputs.
- pdfXmin[setNumber]
 - This function returns the minimum *x* value in the PDF set *setNumber*.

The package pdfParse.m provides the following functions:

- pdfFamilyParseCTEQ[path, [fileType]]
 - This function reads all the files of type *fileType* in the directory *path* and stores them in memory.
 - The function returns a list of set numbers that can be used to define a list. These set numbers correspond to the listing of the .pds files in *pdfSetList*.
 - The optional input *fileType* has a default value of “*.pds”.

- Example: `pdfFamilyParseCTEQ["MyGrids", "ct10 * pds"]` reads all .pds files in the sub-directory "MyGrids" beginning with "ct10" into memory.
- `pdfParseCTEQ[fileName, [verbose]]`
 - This function reads an individual .pds file specified by *fileName* into memory.
 - The function returns a set number that corresponds to the listing of the .pds file in *pdfSetList*.
 - The optional input allows the user to suppress the output of this function by choosing *verbose* to be *False*.

The package `pdfParseLHA.m` provides the following functions:

- `pdfFamilyParseLHA[path, [fileType]]`
 - This function reads all the files of type *fileType* in the directory *path* and stores them in memory.
 - The function returns a list of set numbers that can be used to define a list. These set numbers correspond to the listing of the .dat files in *pdfSetList*.
 - The optional input *fileType* has a default value of "*.dat".
 - Example: `pdfFamilyParseLHA["MyGrids", "ct10 * .dat"]` reads all .dat files in the sub-directory "MyGrids" beginning with "ct10" into memory.
- `pdfParseLHA[fileNameInfo, fileNameData, [verbose]]`
 - This function reads an individual .info file and .data file specified by *fileNameInfo* and *fileNameData*, respectively, into memory.
 - The function returns a set number that corresponds to the listing of the .dat file in *pdfSetList*.
 - Additionally, the function checks that the number and the order of the flavors are the same in both files.
 - The optional input allows the user to suppress the output of this function by choosing *verbose* to be *False*.

The package `pdfErrors.m` provides the following functions:

- `pdfHessianCorrelation[family, list1, list2]`
 - This function returns the correlation between *list1* and *list2* of observables calculated with Hessian PDF error sets in *family*.
 - *Note*: The lists *list1*, *list2* and *family* should all be the same length and correspond to the same PDF sets in memory.
- `pdfHessianError[family, flavor, x, Q, [method]]`
 - This function returns the PDF uncertainty for Hessian PDF error sets in *family*, at given momentum fraction *x* and scale *Q*.
 - The optional input *method* defaults to "sym" for the symmetric error. You may also set this input to "plus" or "minus" for the positive and negative asymmetric errors.
 - *Warning*: The function assumes that the first member of family is the central value PDF set followed by an even number of PDF eigenvector sets.
 - The eigenvector sets should alternate between the plus and minus errors for each of the parameters.
 - `pdfHessianError[family, f[setNumber], [method]]`: Will accept a list or function *f* of sets *setNumber* obtained over a Hessian PDF family.

- pdfLuminosity[setNumber, sqrtS, mX, flavor1, flavor2, [precisionGoal]]
 - This function returns the integrated parton-parton luminosity for collider energy $\sqrt{s} = S^{1/2}$, particle mass mX , and PDF flavors $flavor1$ and $flavor2$, for the set $setNumber$.
 - The numerical integral is performed with the precision goal in the optional parameter $precisionGoal$, which has a default value of $precisionGoal = 3$.
 - The parton luminosity is defined according to Eq.(46) in Campbell, Huston, Stirling, arXiv:hep-ph/0611148v1
- pdfMCCentral[family, flavor, x, Q]
 - This function returns the central value for a Monte Carlo PDF replica set in the list $family$.
 - pdfMCCentral[family, f[setNumber]]: Will accept a list or function f of sets $setNumber$ obtained over a Monte Carlo PDF replica family.
 - *Warning*: The function assumes that the first member of family is the central value PDF set and it is followed by an even number of PDF replica sets.
- pdfMCCorrelation[family, list1, list2]
 - This function returns the correlation between $list1$ and $list2$ of observables calculated with Monte Carlo PDF error sets in $family$.
 - *Note*: The lists $list1$, $list2$ and $family$ should all be the same length and correspond to the same PDF sets in memory.
- pdfMCErrors[family, flavor, x, Q]
 - This function returns the symmetric PDF uncertainty for Monte Carlo PDF error sets in $family$.
 - pdfMCErrors[family, f[setNumber]]: Will accept a list or function f of sets $setNumber$ obtained over a Monte Carlo PDF replica family.
 - *Warning*: The function assumes that the first member of family is the central value PDF set followed by an even number of PDF replica sets.