
Simple Demo file to read Pion PDFs with LHAPDF Tables: xFitter Pion sample table.

Demonstration to show how to use LHAPDF PION tables with ManeParse
This is a simpler demo file that just shows the basics of loading the PDFs.

Please cite:

ManeParse : A Mathematica reader for Parton Distribution Functions
D.B.Clark, E.Godat, F.I.Olness
Published in : Comput.Phys.Commun .216 (2017) 126 - 137
e - Print : 1605.08012[hep - ph]

BUG: (April 2021: Version 4.0)

In the xxx.info file, input fields (such as "SetDesc:") CANNOT be split across multiple lines.
While this is valid YAML, ManeParse reads a single line (record) at a time.
The "patch" for this is simply remove \newlines.

E.g.,

SetDesc: "This will BREAK \cr
the ManeParse Program"

SetDesc: "This will NOT break the ManeParse Program"

GRVPI1: BUG:

The up and up-bar PDFs are switched in the LHAPDF file.
(See example below.)

Fred : 13 June 2020

```
In[1]:= Clear["Global`*"];  
  
In[2]:= $Version  
Out[2]= 12.1.0 for Linux x86 (64-bit) (March 14, 2020)
```

Set Directory

This example notebook is written with relative directories and is intended to be run within the folder extracted from the tarball.

```
In[3]:= (* This just drops the leading path
info to make the list of files easier to read *)
dropPath = Take[(FileNameSplit /@ #) // Transpose, -1][[1]] &;

In[4]:= NotebookDirectory [];
here = NotebookDirectory []

Out[5]= /home/olness/Dropbox/mp/ManeParse5_DEMO /FOR WEB/PION_DEMO/

In[6]:= (* If there is a problem with the Mathematica working directory,
you can enter it manually here *)
SetDirectory[here]

Out[6]= /home/olness/Dropbox/mp/ManeParse5_DEMO /FOR WEB/PION_DEMO

In[7]:= (* This shows what files should be in this main directory *)
FileNames["*", here] // dropPath

Out[7]= {LHAPDF, MP_packages, testPION_v03.nb}
```

Setup Other Directories

```
In[8]:= dirPackages = here <> "MP_packages ";
FileNames["*", dirPackages] // dropPath

Out[9]= {pdfCalc.m, pdfErrors.m, pdfParseCTEQ.m,
pdfParseLHA.m, README_V05.TXT, README_V05.TXT~}

In[10]:= dirFilesLHA = "/usr/local/share/LHAPDF";
dirFilesLHA = here <> "/LHAPDF";
dirList = FileNames["*", dirFilesLHA];
dirList // dropPath

Out[13]= {GRVPI1, xFitterPI}
```

Load the package

Loading the main package provides many useful functions

```
In[14]:= Get[dirPackages <> "/pdfParseLHA .m"]
```

```
Version : pdfCalc 5.0
Version : ManeParse 5.0: April 2021
- Required Package : pdfCalc --Loaded -
```

```
=====
```

```
- pdfParseLHA -
Version : 5.0: April 2021
Authors : E.J. Godat , D.B. Clark & F.I. Olness
```

```
Please cite: *****
http://ncteq.hepforge.org/code/pdf.html
```

```
For a list of available commands , enter : ?pdf*
```

```
=====
```

```
In[15]:= Get[dirPackages <> "/pdfParseCTEQ .m"]
```

```
=====
```

```
- pdfParseCTEQ -
Version : 5.0: April 2021
Authors : D.B. Clark , E.J. Godat & F.I. Olness
```

```
Please cite: *****
http://ncteq.hepforge.org/code/pdf.html
```

```
For a list of available commands , enter : ?pdf*
```

```
=====
```

```
In[16]:= Get[dirPackages <> "/pdfErrors.m"]
```

```
=====
```

```
- pdfErrors -
```

```
Version : 5.0; April 2021
```

```
Authors : D.B. Clark, E.J. Godat & F.I. Olness
```

```
Please cite: *****
```

```
http://ncteq.hepforge.org/code/pdf.html
```

```
For a list of available commands, enter: ?pdf*
```

```
=====
```

All functions begin with 'pdf'. To obtain a list of available functions, type the command '?pdf*'.

```
In[17]:= ? pdf*
```

```
Out[17]=
```

▼ pdfCalc`

pdfAlphaS

pdfFlavor

pdfFunction

pdfFunctionX

pdfGetInfo

pdfGetQlist

pdfGetXlist

pdfLowFunction

pdfLuminosity

pdfNumQpartition

pdfReset

pdfSetInterpolator

pdfSetList

pdfSetListDisplay

pdfSetXpower

pdfXmin

▼ pdfErrors`

pdfError

pdfFamilyFunction

pdfHessianCorrelation

pdfHessianError

pdfMCCentral

pdfMCCentralInterval

pdfMCCorrelation

pdfMCErrors

▼ pdfParseCTEQ`

pdfFamilyParseCTEQ

pdfParseCTEQ

▼ pdfParseLHA`

pdfFamilyParseLHA

pdfParseLHA

Individual file manipulation

In[18]:= **pdfReset []**

Default Mathematica interpolator will be used.

All internal variables have been reset.

Individual files in either LHA or PDS format can be parsed using the functions loaded from the packages. Here we demonstrate the LHA parsing function

In[19]:= **? pdfParseLHA**

Symbol

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* .

Out[19]= 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* .

In[20]:= **fileList = dirList ;**

fileList // dropPath

Out[21]= {GRVPI1 , xFitterPI}

In[22]:= **FileNames["*", dirList[[1]] // dropPath**

Out[22]= {GRVPI1_0000.dat , GRVPI1.info}

In[23]:= **{fileDat , fileInfo} = FileNames["*", dirList[[1]]**

Out[23]= {/home/olness/Dropbox/mp/ManeParse5_DEMO /FOR
WEB/PION_DEMO//LHAPDF/GRVPI1/GRVPI1_0000.dat ,
/home/olness/Dropbox/mp/ManeParse5_DEMO /FOR WEB/PION_DEMO//LHAPDF/GRVPI1/GRVPI1.info}

In[24]:= **fileDat // FullForm**

Out[24]//FullForm=
"/home/olness/Dropbox/mp/ManeParse5_DEMO /FOR
WEB/PION_DEMO//LHAPDF/GRVPI1/GRVPI1_0000.dat"

```
In[25]:= fileInfo // FullForm
```

```
Out[25]//FullForm=
```

```
"/home/olness/Dropbox/mp/ManeParse5_DEMO /FOR WEB/PION_DEMO //LHAPDF /GRVPI1 /GRVPI1 .info"
```

```
In[26]:= pdfParseLHA [fileInfo , fileDat]
```

```
Successfully read
```

```
/home/olness/Dropbox /mp/ManeParse5_DEMO /FOR WEB/PION_DEMO //LHAPDF /GRVPI1 /GRVPI1 .info .
```

```
Successfully read
```

```
/home/olness/Dropbox /mp/ManeParse5_DEMO /FOR WEB/PION_DEMO //LHAPDF /GRVPI1 /GRVPI1_0000 .dat .
```

```
Out[26]= 1
```

```
In[27]:= ? pdfFamilyParseLHA
```

Symbol

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* .

```
Out[27]= The optional input fileType has a default value of "*.dat".
```

Example :

pdfFamilyParseLHA ["MyGrids ","ct10*.dat"] reads all .dat files in the subdirectory "MyGrids " beginning with "ct10" into memory .

```
In[28]:= dirList [[2]]
```

```
Out[28]= /home/olness/Dropbox/mp/ManeParse5_DEMO /FOR WEB/PION_DEMO //LHAPDF/xFitterPI
```

```
In[29]:= (* CAUTION: For this demo file,
only 3 error PDF tables are included to save space *)
pdfFamilyParseLHA [dirList[[2]]]
```

... **Part**: Part 2 of { NLO negative pion fitted to Drell-Yan data of experiments NA10 and E615 and to prompt photon production data of experiment WA70 .} does not exist .

... **Part**: Part 2 of { Nuclear PDF set nCTEQ15FullNuc_184_74 was used for tungsten target .} does not exist .

... **Part**: Part 2 of { Error members 1 to 7 are symmetric Hessian error eigenvectors .} does not exist .

... **General** : Further output of Part::partw will be suppressed during this calculation .

Successfully read

/home/olness/Dropbox/mp/ManeParse5_DEMO /FOR WEB/PION_DEMO /LHAPDF/xFitterPI/xFitterPI .info .

Included 14 files in the PDF family .

```
Out[29]:= {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}
```

```
In[30]:= pdfSetListDisplay []
```

Set Number	File Name	Max Flavors	Valance Flavors
1	/home/olness/Dropbox/mp/ManeParse5_DEMO /FOR WEB/PION_DEMO//LHAPDF/GRVPI1/GRVPI1_0000. dat	5	n/a
2	/home/olness/Dropbox/mp/ManeParse5_DEMO /FOR WEB/PION_DEMO/LHAPDF/xFitterPI/xFitterPI_0000. dat	NumFlavors	n/a
3	/home/olness/Dropbox/mp/ManeParse5_DEMO /FOR WEB/PION_DEMO/LHAPDF/xFitterPI/xFitterPI_0001. dat	NumFlavors	n/a
4	/home/olness/Dropbox/mp/ManeParse5_DEMO /FOR WEB/PION_DEMO/LHAPDF/xFitterPI/xFitterPI_0002. dat	NumFlavors	n/a
5	/home/olness/Dropbox/mp/ManeParse5_DEMO /FOR WEB/PION_DEMO/LHAPDF/xFitterPI/xFitterPI_0003. dat	NumFlavors	n/a
6	/home/olness/Dropbox/mp/ManeParse5_DEMO /FOR WEB/PION_DEMO/LHAPDF/xFitterPI/xFitterPI_0004. dat	NumFlavors	n/a
7	/home/olness/Dropbox/mp/ManeParse5_DEMO /FOR WEB/PION_DEMO/LHAPDF/xFitterPI/xFitterPI_0005. dat	NumFlavors	n/a
8	/home/olness/Dropbox/mp/ManeParse5_DEMO /FOR WEB/PION_DEMO/LHAPDF/xFitterPI/xFitterPI_0006. dat	NumFlavors	n/a

```
Out[30]:=
```

9	/home/olness/Dropbox/mp/ManeParse5_DEMO/FORWEB/PION_DEMO/LHAPDF/xFitterPI/xFitterPI_0007.dat	NumFlavors	n/a
10	/home/olness/Dropbox/mp/ManeParse5_DEMO/FORWEB/PION_DEMO/LHAPDF/xFitterPI/xFitterPI_0008.dat	NumFlavors	n/a
11	/home/olness/Dropbox/mp/ManeParse5_DEMO/FORWEB/PION_DEMO/LHAPDF/xFitterPI/xFitterPI_0009.dat	NumFlavors	n/a
12	/home/olness/Dropbox/mp/ManeParse5_DEMO/FORWEB/PION_DEMO/LHAPDF/xFitterPI/xFitterPI_0010.dat	NumFlavors	n/a
13	/home/olness/Dropbox/mp/ManeParse5_DEMO/FORWEB/PION_DEMO/LHAPDF/xFitterPI/xFitterPI_0011.dat	NumFlavors	n/a
14	/home/olness/Dropbox/mp/ManeParse5_DEMO/FORWEB/PION_DEMO/LHAPDF/xFitterPI/xFitterPI_0012.dat	NumFlavors	n/a
15	/home/olness/Dropbox/mp/ManeParse5_DEMO/FORWEB/PION_DEMO/LHAPDF/xFitterPI/xFitterPI_0013.dat	NumFlavors	n/a

Test PDFs

The function “pdf” is left to be defined by the user. Access to the PDF of the set is given by `pdfFunction`. The function has the canonical form: `pdfFunction [setNumber , flavorNumber , x, Q]`. If the function is not defined, `pdfFunction` returns `NULL`.

In[31]:= **? pdfFunction**

Symbol

`pdfFunction [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` .

Out[31]=

Warning : The results of this function are only reliable between the maximum and minimum values of `x` and `Q` in the `.pds/.dat` file.


```
In[32]:= (* Note, if the flavor is undefined, it will return zero *)
{pdfFunction[1, 1, 0.1, 10],
 pdfFunction[2, 1, 0.1, 10]}
```

```
Out[32]= {2.77328, 3.72318}
```

```
In[33]:= Clear[pdf]
pdf[iset_?IntegerQ, ipart_?IntegerQ, x_?NumericQ, q_?NumericQ] :=
  pdfFunction[iset, ipart, x, q]
SetAttributes[pdf, Listable]
```

SumRule

```
In[36]:= q0 = 1.3;
NIntegrate[Sum[x pdf[1, iparton, x, q0], {iparton, -5, 5}], {x, 0, 1}]
NIntegrate[Sum[x pdf[2, iparton, x, q0], {iparton, -5, 5}], {x, 0, 1}]
```

```
Out[37]= 1.01026
```

```
Out[38]= 1.00006
```

GRVPi1: {u,ub} are swapped {+2 \longleftrightarrow -2}

```
In[39]:= Clear[mom]
Off[NIntegrate::izero];
mom[iparton_, iset_, q0_ : 1.3] := NIntegrate[x pdf[iset, iparton, x, q0], {x, 10-4, 1}]
SetAttributes[mom, Listable]
```

```
In[43]:= grvMom = mom[Range[-5, 5], 1] * 100 // Round;
xfitMom = mom[Range[-5, 5], 2] * 100 // Round;
```

```
In[45]:= tmp = Join[{pdfFlavor /@ Range[-5, 5], Range[-5, 5], grvMom, xfitMom}] // Transpose //
Reverse ;
Prepend[tmp, {"Flavor", "index", "GRVPi1", "xFitter"}] // Grid[#, Frame → All] &
```

Out[46]=

Flavor	index	GRVPi1	xFitter
bottom	5	0	0
charm	4	0	0
strange	3	1	4
up	2	23	4
down	1	23	32
gluon	0	50	19
dbar	-1	1	4
ubar	-2	1	32
sbar	-3	1	4
cbar	-4	0	0
bbar	-5	0	0

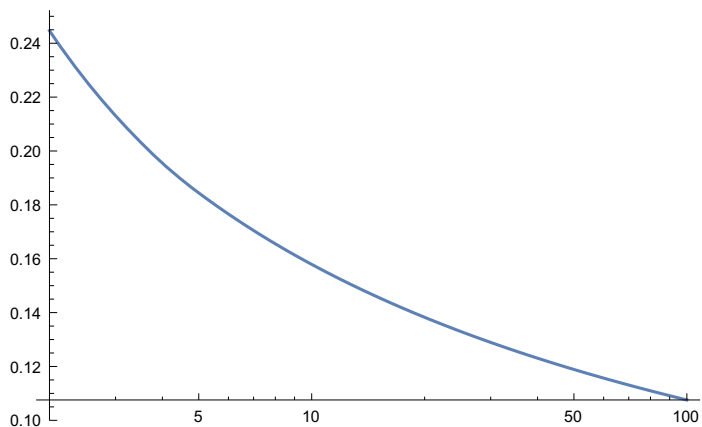
Example: Alpha-S

```
In[47]:= LogLinearPlot[pdfAlphaS[1, q], {q, 2, 100}]
```

Created pdfAlphaS for iSet = 1

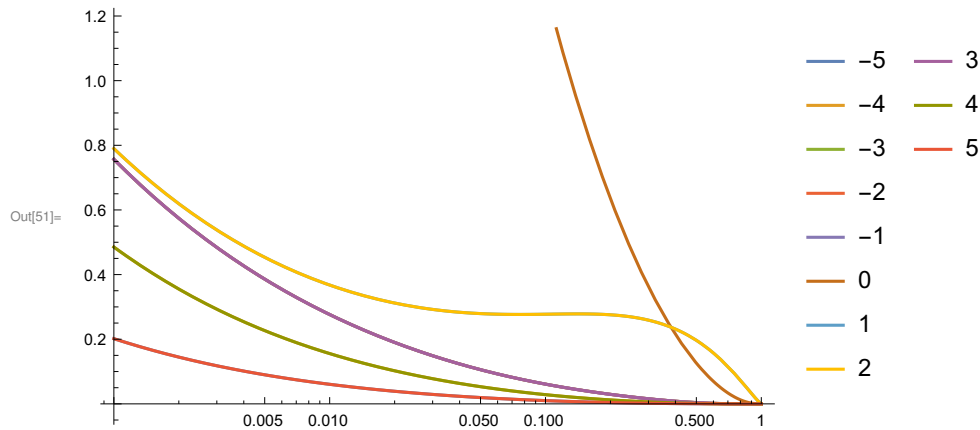
1 has 1 sub-grid

Out[47]=

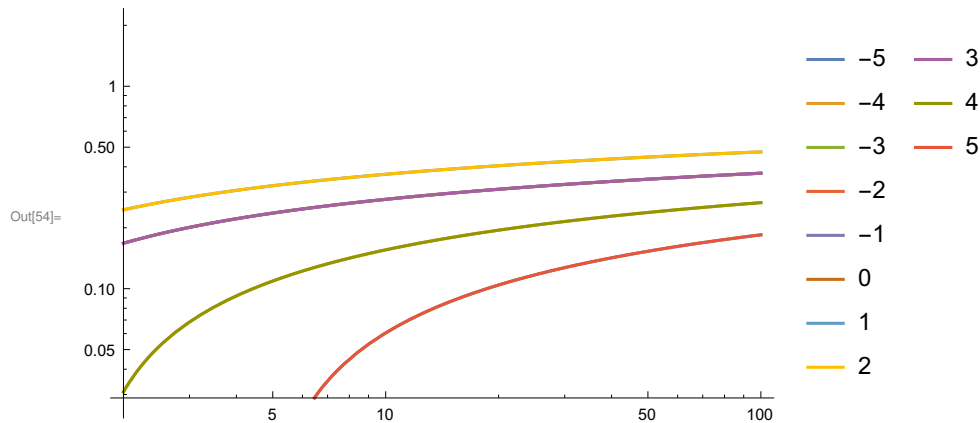


Example: Plotting Single Functions

```
In[48]:= Clear[x];
q0 = 10. ;
flavors = Range[-5, 5];
LogLinearPlot[x pdf[1, flavors, x, q0] // Evaluate, {x, 10-3, 1}, PlotLegends → flavors]
```



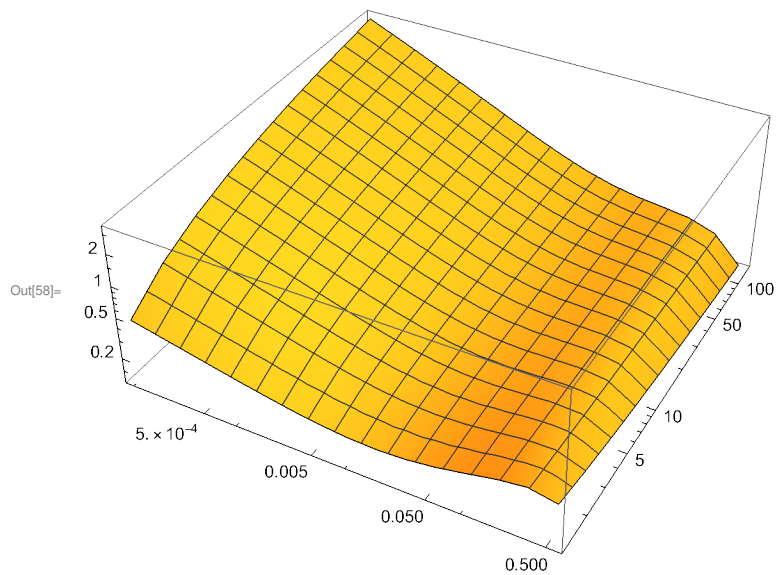
```
In[52]:= Clear[q];
x0 = 0.01 ;
LogLogPlot[x0 pdf[1, flavors, x0, q] // Evaluate, {q, 2, 100}, PlotLegends → flavors]
```



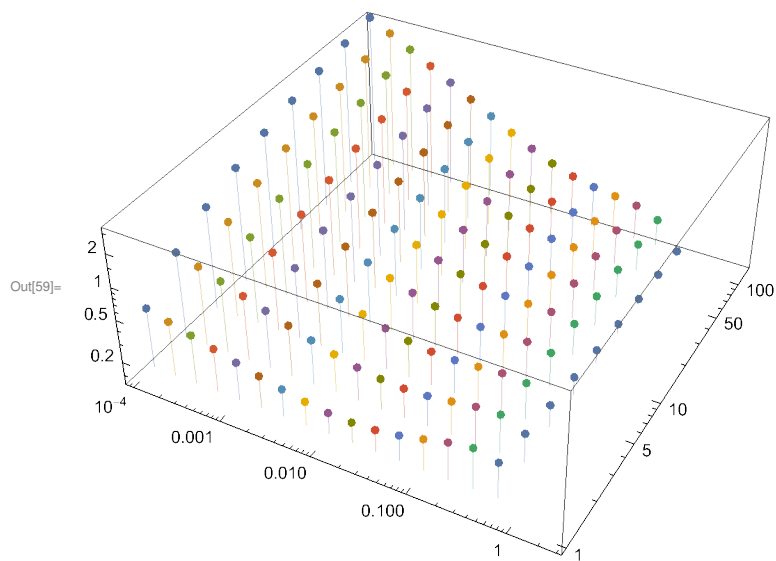
Example: 3D Plots

```
In[55]:= xlist = Table[10.^i, {i, -4, 0, 1/4}];
qlist = Table[1.3 * 10.^i, {i, 0, 2, 1/4}];
data = Table[{x = xlist[[i]], q = qlist[[j]], x pdf[1, 1, x, q]},
  {i, 1, Length[xlist]}, {j, 1, Length[qlist]}];
```

```
In[58]:= ListPlot3D[data // Flatten[#, 1] &, ScalingFunctions → {"Log", "Log", "Log"}]
```



```
In[59]:= ListPointPlot3D[data, ScalingFunctions → {"Log", "Log", "Log"},  
Filling → Bottom]
```



Valence, Sea, Glue
{2(d-dbar), 6 u, g}

```
In[60]:= Clear[grv, xfit];
```

```

In[61]:= grv[x_, q_] := Module[{valence, sea, glue, upquark, iset0 = 1 (* GRVPi1 *)},
  upquark = -2; (* THIS IS A PATCH TO FIX THE SWAPPED COLUMNS *)
  valence = 2 (pdf[iset0, +1, x, q] - pdf[iset0, -1, x, q]);
  sea = 6 (pdf[iset0, upquark, x, q]);
  glue = pdf[iset0, 0, x, q];
  Return[{valence, sea, glue}]
]
grv[0.1, 10.]
Out[62]= {4.32637, 3.66061, 12.853}

In[63]:= xfit[x_, q_] := Module[{valence, sea, glue, upquark, iset0 = 2 (* xFitter *)},
  upquark = +2; (* No swap needed *)
  valence = 2 (pdf[iset0, +1, x, q] - pdf[iset0, -1, x, q]);
  sea = 6 (pdf[iset0, upquark, x, q]);
  glue = pdf[iset0, 0, x, q];
  Return[{valence, sea, glue}]
]
xfit[0.1, 10.]
Out[64]= {4.55502, 8.67406, 9.39553}

```

Plots:

```

In[65]:= q0 = 2.0;
part = 1; (* valence *)
names = {"valence", "sea", "gluon"};
doPlot[part_] := Plot[{x * grv[x, q0][[part]], x * xfit[x, q0][[part]]}, {x, 0, 1},
  PlotStyle -> {Red, Blue},
  PlotLabel -> names[[part]],
  PlotLegends -> {"GRVPi1", "xFitter"}
]

```

```
In[69]:= Do[  
  doPlot[part] // Print,  
  {part, 1, 3, 1}]
```

