Simple Demo file to LHAPDF Tables:

Demonstration to show how to use LHAPDF PDF 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]

Fred : 26 May 2020

```
In[ ]:= Clear["Global`*"];
```

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

```
In[ ]:= NotebookDirectory[];
        here = NotebookDirectory[]
Out[ ]= /home/olness/Downloads/SF_DEMO_v01/PDF_DEMO_v01/
```

```
In[ ]:= (* If there is a problem with the Mathematica working directory,
        you can enter it manually here *)
        SetDirectory[here]
Out[ ]= /home/olness/Downloads/SF_DEMO_v01/PDF_DEMO_v01
```

```
In[ ]:= (* This shows what files should be in this main directory *)
        FileNames["*", here] // dropPath
Out[ ]= {LHAPDF, MP_packages, testPDF_v01.nb}
```

Setup Other Directories

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

Out[ ]= {pdfCalc .m, pdfErrors .m, pdfParseCTEQ .m, pdfParseLHA .m}
```

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

Out[ ]= {nCTEQ15FullNuc_ 1_1, nCTEQ15FullNuc_ 208_82}
```

Load the package

Loading the main package provides many useful functions

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

```
  - Required  Package : pdfCalc  --Loaded  -


=== === === === === === === === === === === === === === === === === === === === === ===


  - pdfParseLHA  -

Version :  1.0

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

=== === === === === === === === === === === === === === === === === === === === ===


 - pdfParseCTEQ  -

Version :  1.0

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

=== === === === === === === === === === === === === === === === === === === === ===


 - pdfErrors  -

Version :  1.0

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[ ]:= ? pdf*
```

```
Out[ ]=
⌄ pdfCalc`

                                                    pdfNumQ⋱     pdfSetInte⋱   pdfSetList⋱
pdfAlphaS    pdfFunction    pdfGetQlist    pdfLowFu⋱   partitio⋱   rpolator      Display      pdfXmin
                                             nction      n

pdfFlavor    pdfGetInfo     pdfGetXlist    pdfLumin⋱   pdfReset    pdfSetList    pdfSetXpo⋱
                                             osity                                  wer

⌄ pdfErrors`
pdfError               pdfHessianCorrelation    pdfMCCentral            pdfMCCorrelation
pdfFamilyFunction      pdfHessianError          pdfMCCentralInterval    pdfMCError

⌄ pdfParseCTEQ`
pdfFamilyParseCTEQ                              pdfParseCTEQ

⌄ pdfParseLHA`
pdfFamilyParseLHA                               pdfParseLHA
```

Individual file manipulation

```
In[ ]:= 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[ • ]:= ? pdfParseLHA
```

Symbol

Out[•]=

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 supress the output of this function by choosing *verbose* to be *False* .

```
In[ • ]:= fileList = dirList // First;
          fileList // dropPath

Out[ • ]= /home/olness/Downloads/SF_DEMO_v01/PDF_DEMO_v01//LHAPDF/nCTEQ15FullNuc_ 1_1

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

Out[ • ]= {nCTEQ15FullNuc_ 1_1_0000.dat, nCTEQ15FullNuc_ 1_1.info}

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

Out[ • ]= {/home/olness/Downloads/SF_DEMO_v01/PDF_DEMO_v01//LHAPDF/nCTEQ15FullNuc_ 1_1/
             nCTEQ15FullNuc_ 1_1_0000.dat,
           /home/olness/Downloads/SF_DEMO_v01/PDF_DEMO_v01//LHAPDF/nCTEQ15FullNuc_ 1_1/
             nCTEQ15FullNuc_ 1_1.info}

In[ • ]:= pdfParseLHA[fileInfo, fileDat]

          Successfully read
           /home/olness/Downloads/SF_DEMO_v01/PDF_DEMO_v01//LHAPDF/nCTEQ15FullNuc_ 1_1/nCTEQ15FullNuc_ 1_1.
             info.

          Successfully read
           /home/olness/Downloads/SF_DEMO_v01/PDF_DEMO_v01//LHAPDF/nCTEQ15FullNuc_ 1_1/nCTEQ15FullNuc_ 1_1_
             0000.dat.

Out[ • ]= 1
```

```
In[ ]:= ? 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[ ]= 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[ ]:= dirList[[2]]
Out[ ]= /home/olness/Downloads/SF_DEMO_v01/PDF_DEMO_v01//LHAPDF/nCTEQ15FullNuc_ 208_82
```

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

        Successfully  read
          /home/olness/Downloads/SF_DEMO_v01/PDF_DEMO_v01/LHAPDF/nCTEQ15FullNuc_ 208_82/nCTEQ15FullNuc_
            208_82.info.

        Included 3 files in the PDF family.

Out[ ]= {2, 3, 4}
```

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

Out[ ]=

Set Number	File Name	Max Flavors	Valance Flavors
1	/home/olness/Downloads/SF_DEMO_v01/PDF_DEMO_v01//LHAPDF/nCTEQ15FullNuc_ 1_1/nCTEQ15FullNuc_ 1_1_0000.dat	5	n/a
2	/home/olness/Downloads/SF_DEMO_v01/PDF_DEMO_v01/LHAPDF/nCTEQ15FullNuc_ 208_82/nCTEQ15FullNuc_ 208_82_0000.dat	5	n/a
3	/home/olness/Downloads/SF_DEMO_v01/PDF_DEMO_v01/LHAPDF/nCTEQ15FullNuc_ 208_82/nCTEQ15FullNuc_ 208_82_0001.dat	5	n/a
4	/home/olness/Downloads/SF_DEMO_v01/PDF_DEMO_v01/LHAPDF/nCTEQ15FullNuc_ 208_82/nCTEQ15FullNuc_ 208_82_0002.dat	5	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[ ]:= ? pdfFunction
```

Out[ ]=

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* .

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

Out[ ]= {3.95735 , 5.80936}
```

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

```
In[ ]:= pdfGetInfo[1] // TableForm
```

```
Out[ ]//TableForm=
        SetDesc →  "nCTEQ15 fit
        SetIndex → 102 000
        Authors →  K. Kovarik, A. Kusina, T. Jezo, D. B. Clark, C. Keppel, F. Lyonnet, J. G.
        Reference →  arXiv
        Format →  lhagrid1
        DataVersion → 1
        NumMembers → 1
        Particle → 2212
        Flavors → {-5, -4, -3, -2, -1, 1, 2, 3, 4, 5, 21}
        OrderQCD → 1
        FlavorScheme →  variable
        NumFlavors → 5
        ErrorType →  hessian
        ErrorConfLevel →  90
        XMin → 1/200 000
        XMax → 1.
        QMin → 1.3
        QMax → 10 000.
        MZ → 91.188
        MUp →  0.0
        MDown →  0.0
        MStrange →  0.0
        MCharm →  1.3
        MBottom →  4.5
        MTop →  174.0
        AlphaS_MZ →  1.179973 e-01
        AlphaS_OrderQCD →  1
        AlphaS_Type →  ipol
        AlphaS_Qs → {1.3, 1.49426, 1.73673, 2.0429, 2.43436, 2.94169, 3.60881, 4.5, 5.86604, 7.82
        AlphaS_Vals → {0.396765, 0.361268, 0.330375, 0.303188, 0.27903, 0.257386, 0.237861, 0.22
        AlphaS_Lambda4 →  0.326
        AlphaS_Lambda5 →  0.226
```

```
In[ ]:= flavors = "Flavors" /. pdfGetInfo[1]

Out[ ]= {-5, -4, -3, -2, -1, 1, 2, 3, 4, 5, 21}
```

```
In[ ]:= pdf[1, flavors, 0.1, 10]

Out[ ]= {0.0782868, 0.258414, 0.616414, 0.952443, 1.35407,
         3.95735, 6.50396, 0.616414, 0.258414, 0.0782868, 10.8134}
```

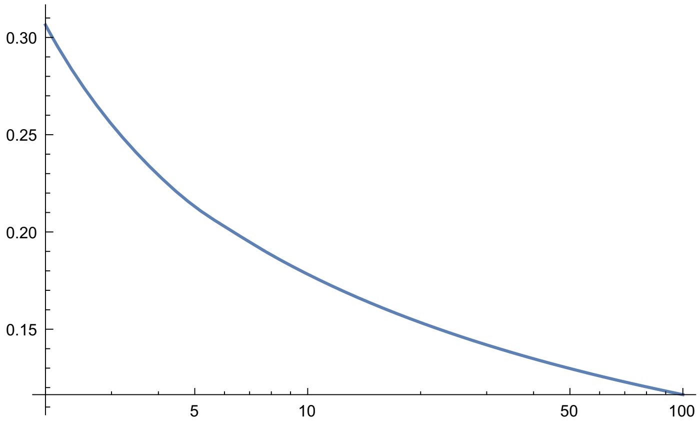
Example: Alpha-S

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

```
Created  pdfAlphaS  for  iSet  =  1

1  has   1  sub-grid
```

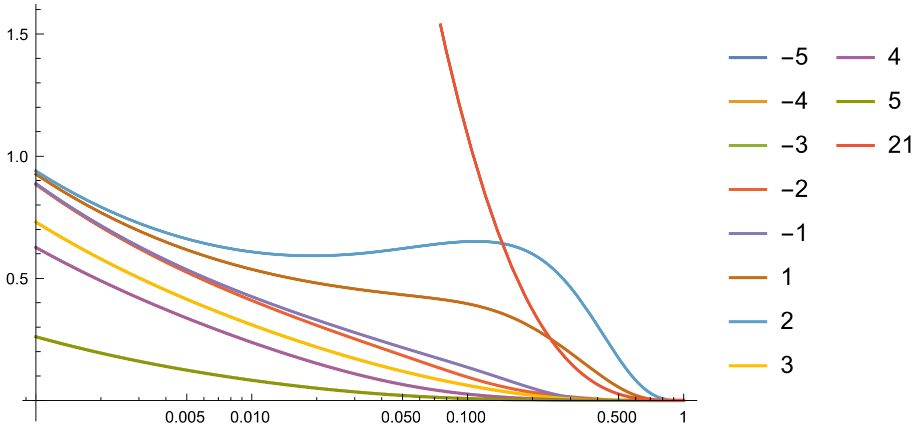
Out[ ]=



Example: Plotting Single Functions

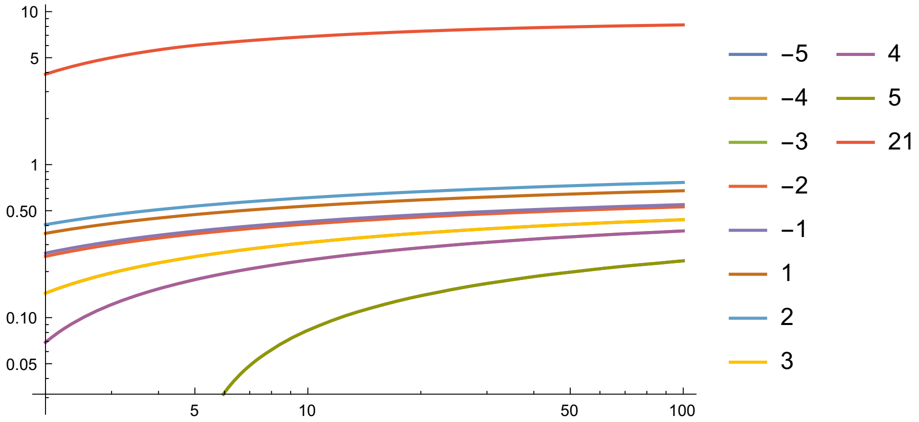
In[ ]:=
```
Clear[x];
q0 = 10. ;
LogLinearPlot[x pdf[1, flavors, x, q0] // Evaluate, {x, 10^-3, 1}, PlotLegends → flavors]
```

Out[ ]=



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

Out[∘]=

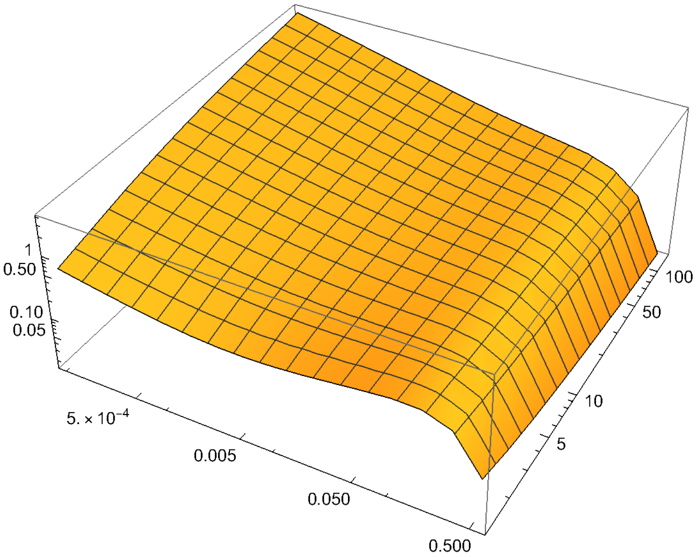


Example: 3D Plots

```
In[ ∘ ]:= 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]}] ;
```

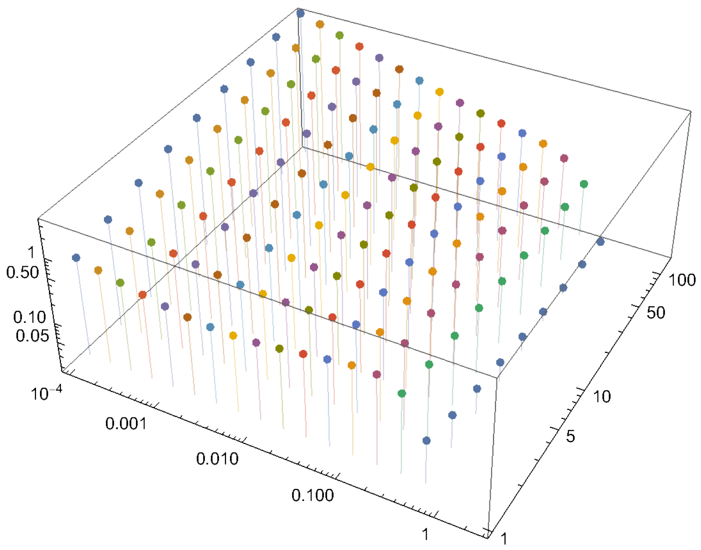
```
In[ ∘ ]:= ListPlot3D[data // Flatten[#, 1] &, ScalingFunctions → {"Log", "Log", "Log"}]
```

Out[∘]=



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

Out[ ]=


1
0.50
0.10
0.05
10^{-4}
0.001
0.010
0.100
1
100
50
10
5
1