

```
Clear["Global`*"];
```

Example file for pdfParseLHA.nb Version 1.0

Version 1.0 beta

15 March 2014

Comments and questions to:

Fred Olness olness@smu.edu

Eric Godat egodat@smu.edu

[NOTE : This is an early "beta" version,

so it can only read a single q – range table. Thus, it works on CTEQ, and NNPDF,

but not in general for the MSTW which uses multiple grids in Q ; this will be updated shortly.]

Set Directory

This example notebook is written with relative directories and is intended to be run within the folder extracted from the tarball. Uncomment and modify the code below to set a different directory for the LHA files.

```
Directory[]
```

```
/home/olness
```

```
(*here=Directory[]*)
```

```
here = "/home/olness/eric/DEMO/"
```

```
/home/olness/eric/DEMO/
```

```
FileNames["*", here <> "LHAPDF/"] // TableForm
```

```
/home/olness/eric/DEMO/LHAPDF/CT10
```

```
/home/olness/eric/DEMO/LHAPDF/CT10nlo
```

```
/home/olness/eric/DEMO/LHAPDF/cteq611
```

```
/home/olness/eric/DEMO/LHAPDF/NNPDF23_nlo_as_0118
```

```
subDir1 = here <> "/LHAPDF/CT10";
subDir2 = here <> "/LHAPDF/CT10nlo";
subDir3 = here <> "/LHAPDF/cteq6l1";
subDir4 = here <> "/LHAPDF/NNPDF23_nlo_as_0118";
```

Load the package

loading the main package provides many useful functions

```
SetDirectory[here]
FileNames["*.m"]
/home/olness/eric/DEMO
{pdfParseLHA.m}

<< pdfParseLHA.m;

=====

- pdfParseLHA -
Version: 1.0
Authors: E.J. Godat, D.B. Clark & F.I. Olness

Please cite: *****

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

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

?pdf*

▼ pdfParseLHA`

pdfAlphaS-LHA	pdfFlavorLHA	pdfGetAlphaValuesLHA	pdfGetQlistLHA	pdfGetValueLHA	pdfParseLHA	pdfSetListLHA
pdfFamilyParserLHA	pdfFunctionLHA	pdfGetInfoLHA	pdfGetTableLHA	pdfGetXlistLHA	pdfResetLHA	pdfXminLHA

Individual file manipulation

```
pdfResetLHA[]
```

All internal variables have been reset.

```
?pdfParseLHA
```

```
pdfParseLHA[fileNameInfo, fileNameData]: This function reads an info file and a data file into
memory and checks that the number and the order of the flavors are the same in both files
```

Read in dir1 [CT10]

```
subDir = subDir1;

datfiles = FileNames["*.dat", subDir]; (* This is a set of LHA PDFs. *)
infofile = FileNames["*.info", subDir]; (* This is the associated info file *)

sample = pdfParseLHA[infofile[[1]], datfiles[[1]]]
Successfully read /home/olness/eric/DEMO//LHAPDF/CT10/CT10.info.
Successfully read /home/olness/eric/DEMO//LHAPDF/CT10/CT10_0000.dat.

sample2 = pdfParseLHA[infofile[[1]], datfiles[[2]]]
Successfully read /home/olness/eric/DEMO//LHAPDF/CT10/CT10.info.
Successfully read /home/olness/eric/DEMO//LHAPDF/CT10/CT10_0001.dat.
```

Read in dir2 [CT10nlo]

```
subDir = subDir2;

datfiles = FileNames["*.dat", subDir]; (* This is a set of LHA PDFs. *)
infofile = FileNames["*.info", subDir]; (* This is the associated info file *)

sample = pdfParseLHA[infofile[[1]], datfiles[[1]]]
Successfully read /home/olness/eric/DEMO//LHAPDF/CT10nlo/CT10nlo.info.
Successfully read /home/olness/eric/DEMO//LHAPDF/CT10nlo/CT10nlo_0000.dat.

sample2 = pdfParseLHA[infofile[[1]], datfiles[[2]]]
Successfully read /home/olness/eric/DEMO//LHAPDF/CT10nlo/CT10nlo.info.
Successfully read /home/olness/eric/DEMO//LHAPDF/CT10nlo/CT10nlo_0001.dat.
```

Read in dir3 [cteq6l1]

```
subDir = subDir3;

datfiles = FileNames["*.dat", subDir]; (* This is a set of LHA PDFs. *)
infofile = FileNames["*.info", subDir]; (* This is the associated info file *)
```

```
sample = pdfParseLHA[infofile[[1]], datfiles[[1]]]
Successfully read /home/olness/eric/DEMO//LHAPDF/cteq611/cteq611.info.
Successfully read /home/olness/eric/DEMO//LHAPDF/cteq611/cteq611_0000.dat.
```

Read in dir4 [NNPDF23]

```
subDir = subDir4;

datfiles = FileNames["*.dat", subDir];(* This is a set of LHA PDFs. *)
infofile = FileNames["*.info", subDir];(* This is the associated info file *)

sample = pdfParseLHA[infofile[[1]], datfiles[[1]]]
Read::readn: Invalid real number found when reading from
/home/olness/eric/DEMO//LHAPDF/NNPDF23_nlo_as_0118/NNPDF23_nlo_as_0118.info. >>
Successfully read
/home/olness/eric/DEMO//LHAPDF/NNPDF23_nlo_as_0118/NNPDF23_nlo_as_0118.info.
Successfully read
/home/olness/eric/DEMO//LHAPDF/NNPDF23_nlo_as_0118/NNPDF23_nlo_as_0118_0000.dat.

sample2 = pdfParseLHA[infofile[[1]], datfiles[[2]]]
Read::readn: Invalid real number found when reading from
/home/olness/eric/DEMO//LHAPDF/NNPDF23_nlo_as_0118/NNPDF23_nlo_as_0118.info. >>
Successfully read
/home/olness/eric/DEMO//LHAPDF/NNPDF23_nlo_as_0118/NNPDF23_nlo_as_0118.info.
Successfully read
/home/olness/eric/DEMO//LHAPDF/NNPDF23_nlo_as_0118/NNPDF23_nlo_as_0118_0001.dat.
```

Calling the pdfSetList variable will give a key to the .dat files in memory. The information is displayed as:
{SetNumber,FileName}

```
pdfSetListLHA // TableForm
1 /home/olness/eric/DEMO//LHAPDF/CT10/CT10_0000.dat
2 /home/olness/eric/DEMO//LHAPDF/CT10/CT10_0001.dat
3 /home/olness/eric/DEMO//LHAPDF/CT10nlo/CT10nlo_0000.dat
4 /home/olness/eric/DEMO//LHAPDF/CT10nlo/CT10nlo_0001.dat
5 /home/olness/eric/DEMO//LHAPDF/cteq611/cteq611_0000.dat
6 /home/olness/eric/DEMO//LHAPDF/NNPDF23_nlo_as_0118/NNPDF23_nlo_as_0118_0000.da
7 /home/olness/eric/DEMO//LHAPDF/NNPDF23_nlo_as_0118/NNPDF23_nlo_as_0118_0001.da
```

Test PDFs

The function “pdf” is left to be defined by the user. Access to the PDF of the set is given by pdfFunctionLHA. The function has the

canonical form: **pdfFunction[setNumber, flavorNumber, x, Q]**. If the function is not defined, pdfFunction returns `NULL`.

?pdfFunctionLHA

pdfFunctionLHA[setNumber,flavor,x,Q]: This function returns the value of the pdf for the .dat file corresponding to *setNumber*, for the given flavor and value of Bjorken *x* and *Q*.

Warning: The results of this function are only reliable between the maximum and minimum values of *x* and *Q* in the .info file.

```
pdfFunctionLHA[1, 1, .1, 10]
```

```
0.396968
```

```
Table[pdfFunctionLHA[iset, 1, .1, 10], {iset, 1, 5, 1}]
```

```
{0.396968, 0.395809, 0.396767, 0.395613, 0.365448}
```

```
Clear[pdf]
```

```
pdf[iset_?IntegerQ, ipart_?IntegerQ, x_?NumericQ, q_?NumericQ] :=  
  pdfFunctionLHA[iset, ipart, x, q]
```

```
pdf[1, 1, 0.1, 10]
```

```
pdf[2, 1, 0.1, 10]
```

```
pdf[-1, 1, 0.1, 10] (* not defined *)
```

```
0.396968
```

```
0.395809
```

Check sum rule:

```
Off[NIntegrate::izero]
```

```
Off[NIntegrate::ncvb]
```

```
q0 = 2.0;
```

```
iset0 = 1;
```

```
tab = Table[NIntegrate[ pdf[iset, ipart, x, q0], {x, 0, 1}],  
  {ipart, -5, 5, 1}, {iset, 1, 7, 1}];
```

```
Plus @@ tab
```

```
{0.999837, 0.999837, 0.99985, 0.999851, 0.999916, 1.00329, 1.00447}
```

```
flavorlist = {};
```

```
For[i = -5, i ≤ 5, i++, AppendTo[flavorlist, pdfFlavorLHA[i]]];
```

```
flavorlist
```

```
{bbar, cbar, sbar, ubar, dbar, gluon, down, up, strange, charm, bottom}
```

```
pdfSetListLHA // TableForm
```

```
1 /home/olness/eric/DEMO//LHAPDF/CT10/CT10_0000.dat
2 /home/olness/eric/DEMO//LHAPDF/CT10/CT10_0001.dat
3 /home/olness/eric/DEMO//LHAPDF/CT10nlo/CT10nlo_0000.dat
4 /home/olness/eric/DEMO//LHAPDF/CT10nlo/CT10nlo_0001.dat
5 /home/olness/eric/DEMO//LHAPDF/cteq611/cteq611_0000.dat
6 /home/olness/eric/DEMO//LHAPDF/NNPDF23_nlo_as_0118/NNPDF23_nlo_as_0118_0000.da
7 /home/olness/eric/DEMO//LHAPDF/NNPDF23_nlo_as_0118/NNPDF23_nlo_as_0118_0001.da
```

```
pdfSets = {
  "CT10-1",
  "CT10-2",
  "CT10nlo-1",
  "CT10nlo-2",
  "cteq611",
  "NNPDF23-0",
  "NNPDF23-1"
};
```

```
Round[ 100 tab] // TableForm[#, TableHeadings → {flavorlist, pdfSets}] &
```

	CT10-1	CT10-2	CT10nlo-1	CT10nlo-2	cteq611	NNPDF23-0	NNPDF23
bbar	0	0	0	0	0	0	0
cbar	0	0	0	0	0	0	0
sbar	2	2	2	2	2	1	2
ubar	3	3	3	3	3	3	3
dbar	4	4	4	4	4	4	4
gluon	42	42	42	42	44	43	42
down	15	15	15	15	15	16	16
up	32	32	32	32	31	31	32
strange	2	2	2	2	2	1	2
charm	0	0	0	0	0	0	0
bottom	0	0	0	0	0	0	0

Example: Plotting Single Functions

First we find the minimum value of x for our pdf family.

```
?pdfXminLHA
```

pdfXminLHA[setNumber]: This function returns the minimum x value in the PDF grid *setNumber*.

```
xMin = pdfXminLHA[1]
```

```
1. × 10-8
```

We will produce plots of $x \cdot \text{pdf}(x, Q)$ for all flavors with the central value in red and the first error set in green. The flavor can be called with the command pdfFlavorLHA[flavor].

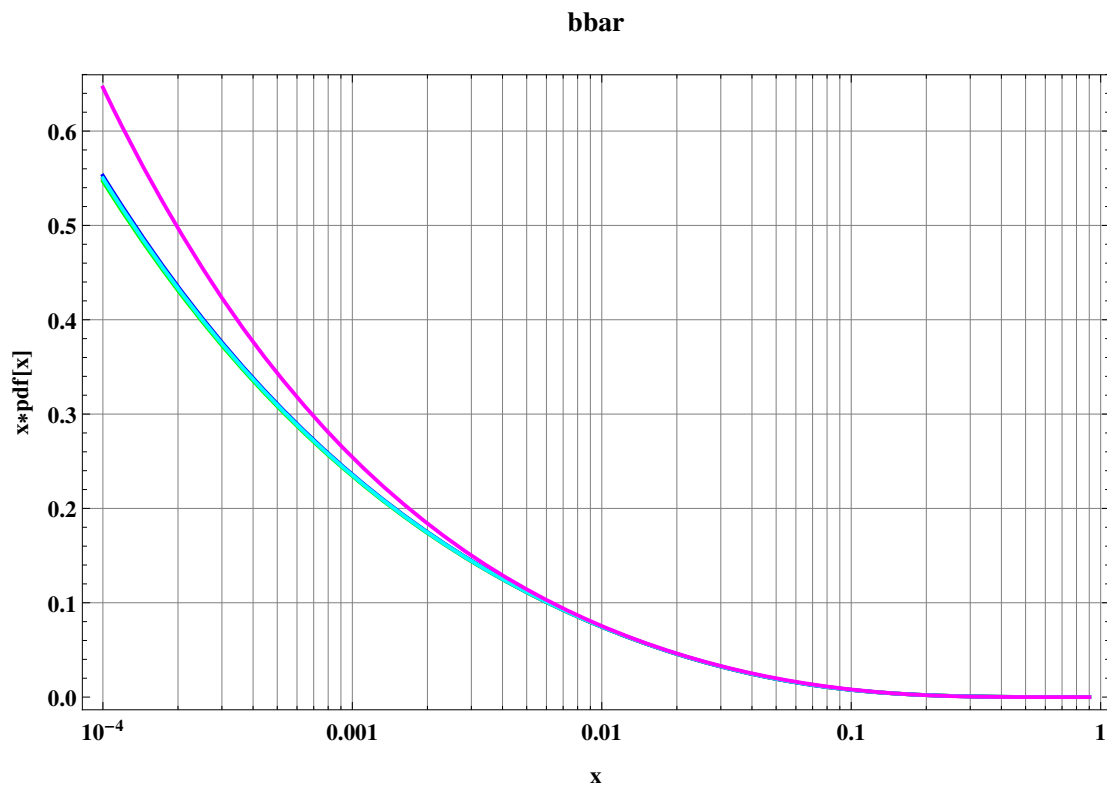
```

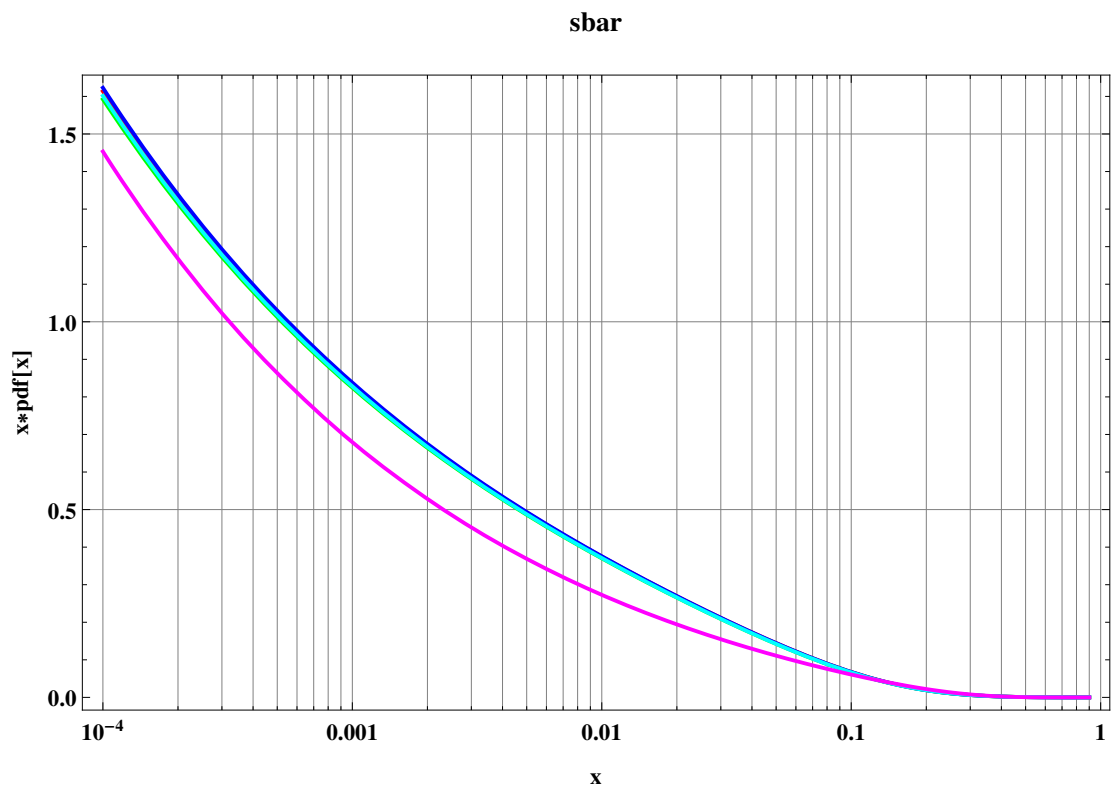
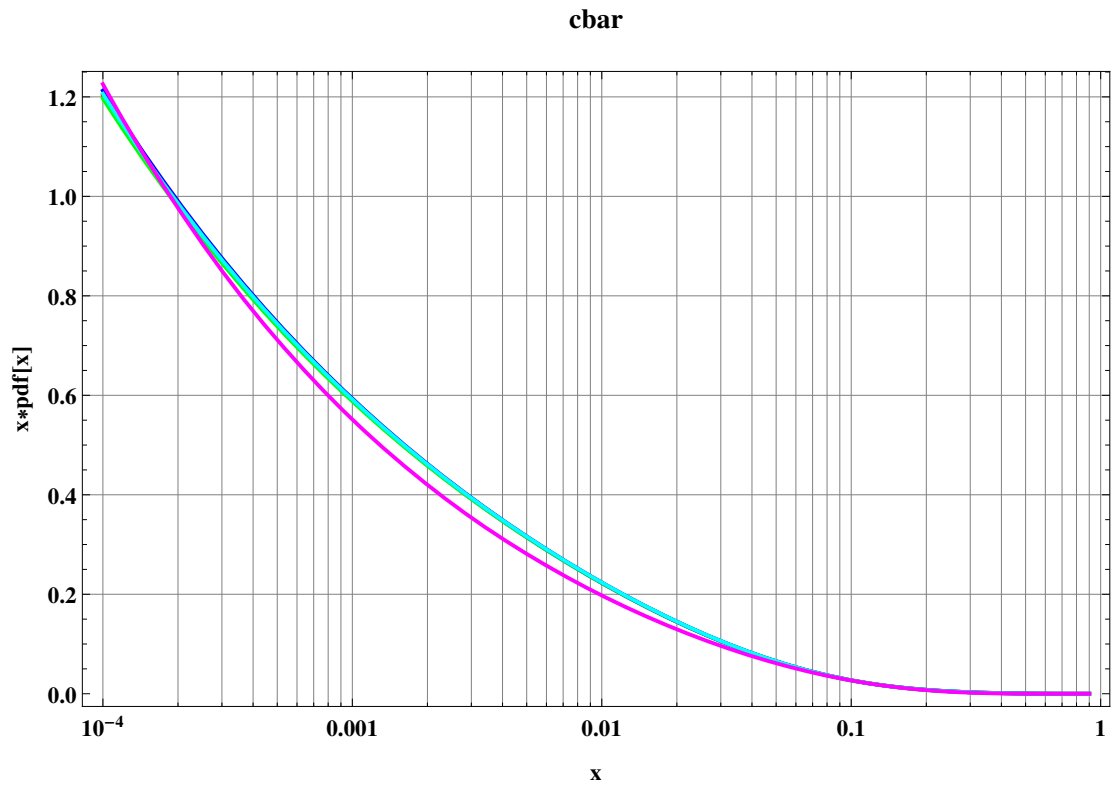
colors = {Red, Green, Blue, Cyan, Magenta, Yellow, Black, Purple, Orange};
colors2 = {#, Thick} & /@ colors;

q0 = 10;

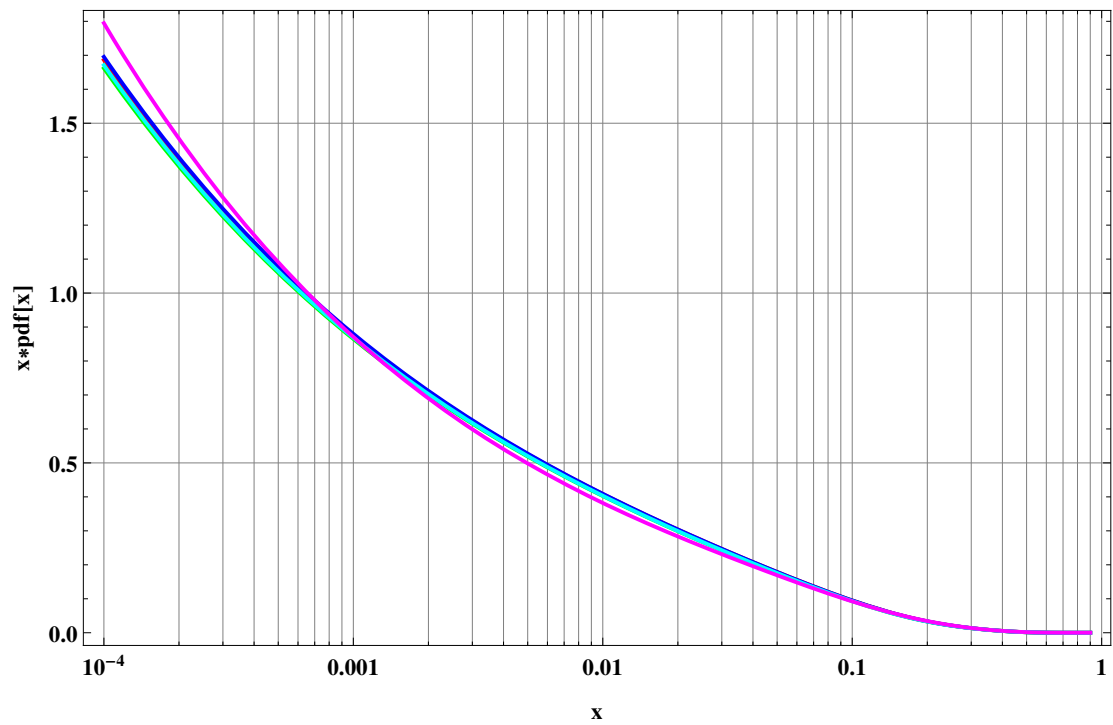
For[i = -5, i ≤ 5, i++,
  LogLinearPlot[
    Table[pdf[iset, i, x, q0], {iset, 1, 5, 1}] // Evaluate, {x, 10.^-4, 0.9},
    PlotStyle → colors2,
    PlotLabel → pdfFlavorLHA[i] ,
    FrameLabel → {"x", "x*pdf[x]"},
    ImageSize → Large,
    PlotRange → All,
    Frame → True,
    BaseStyle → {FontWeight → "Bold", FontSize → 12},
    GridLines → Automatic
  ] // Print
]

```

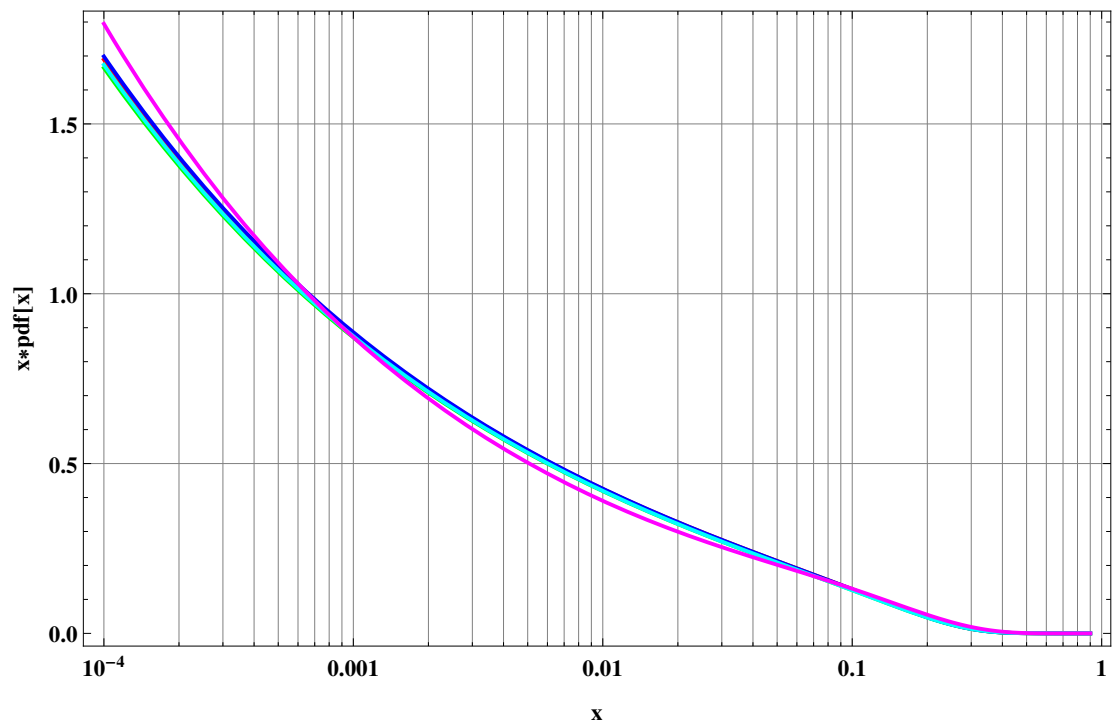


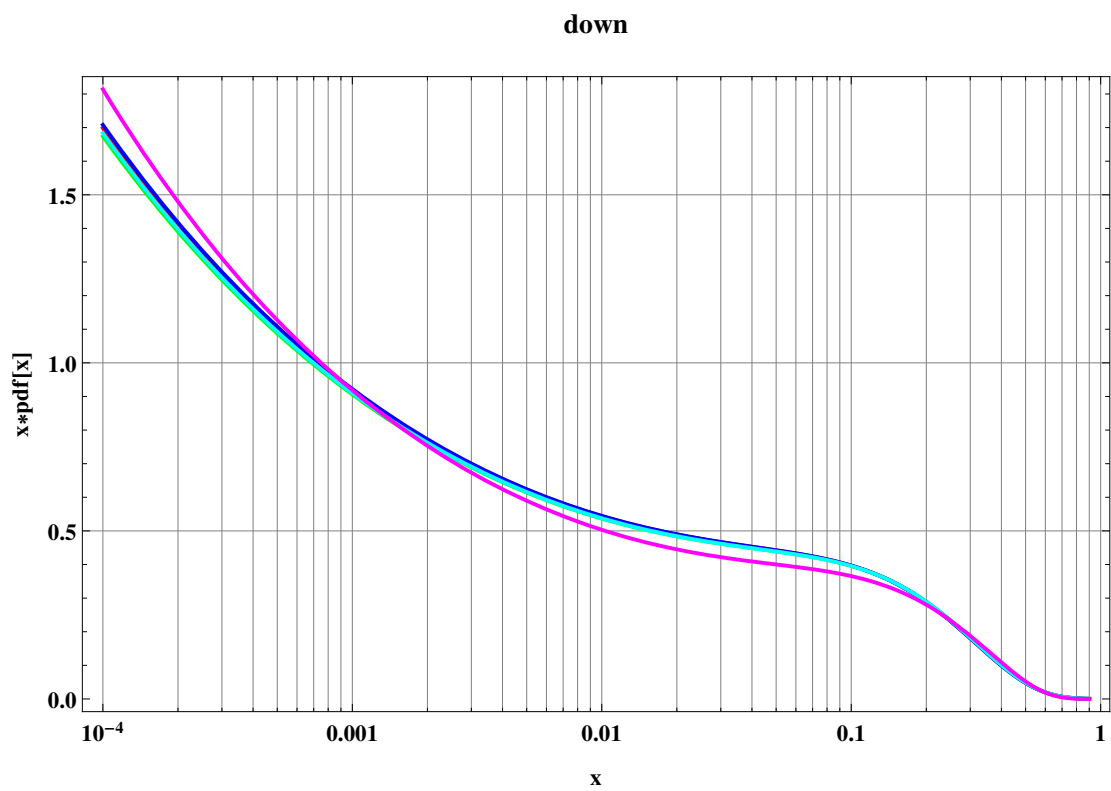
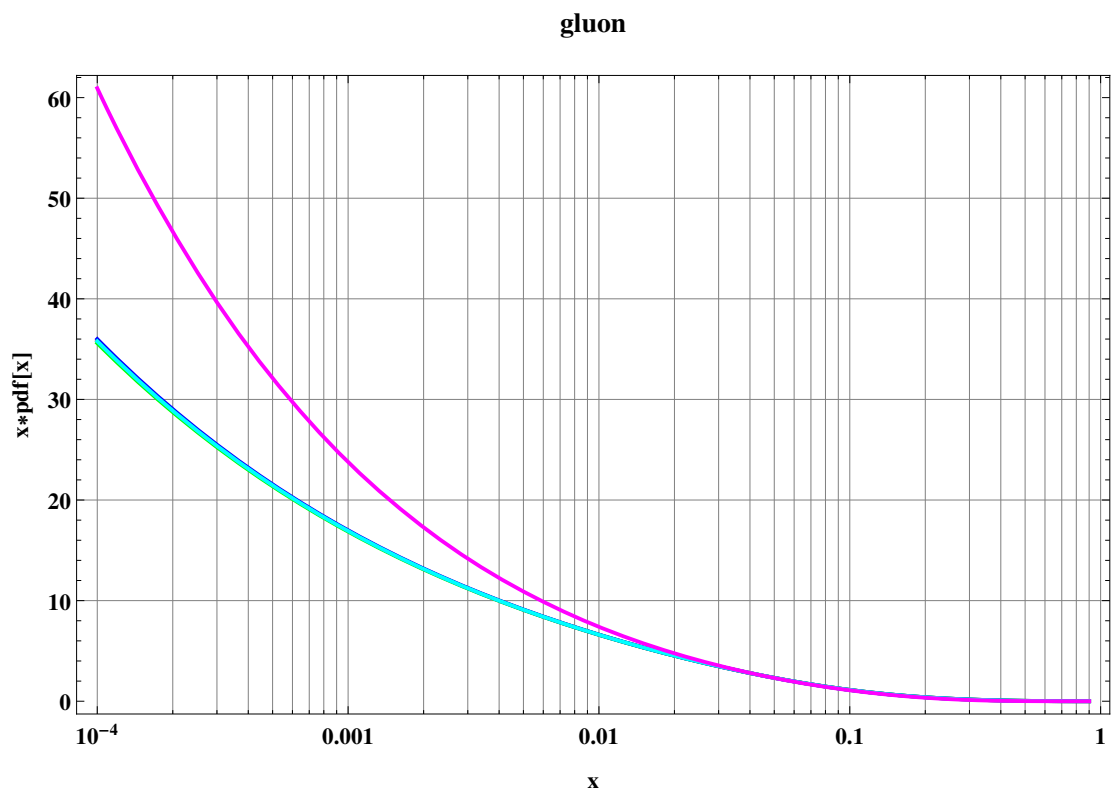


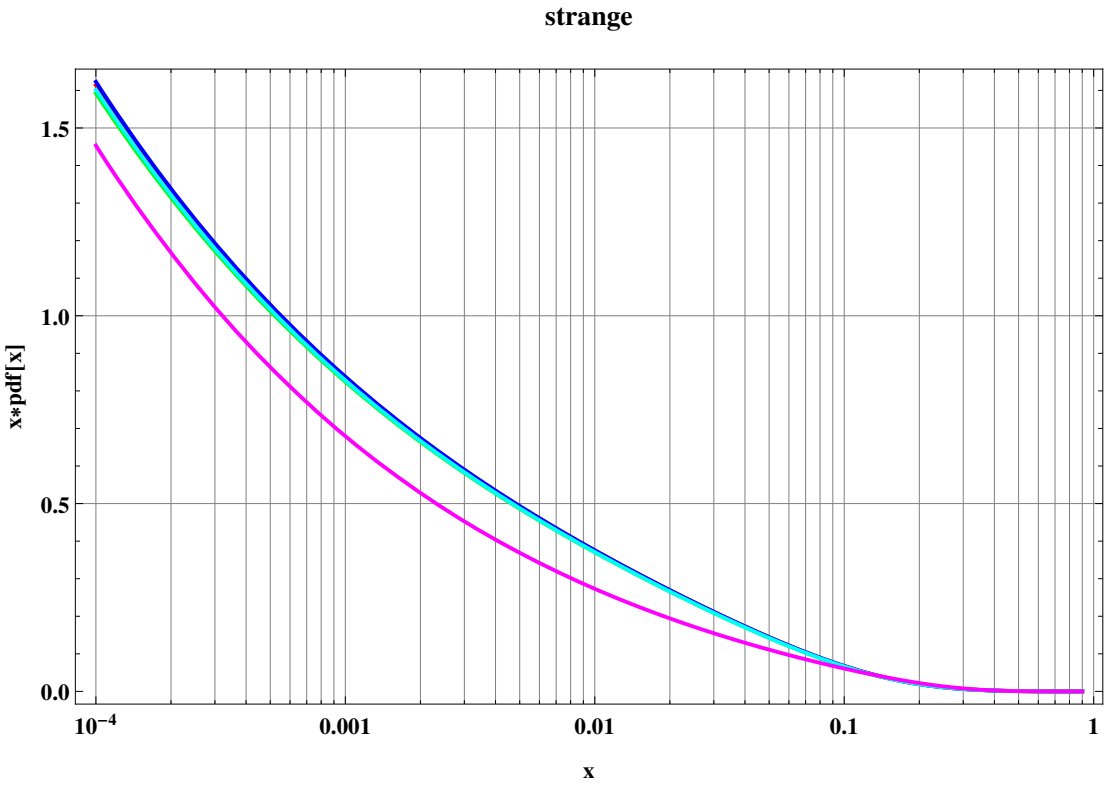
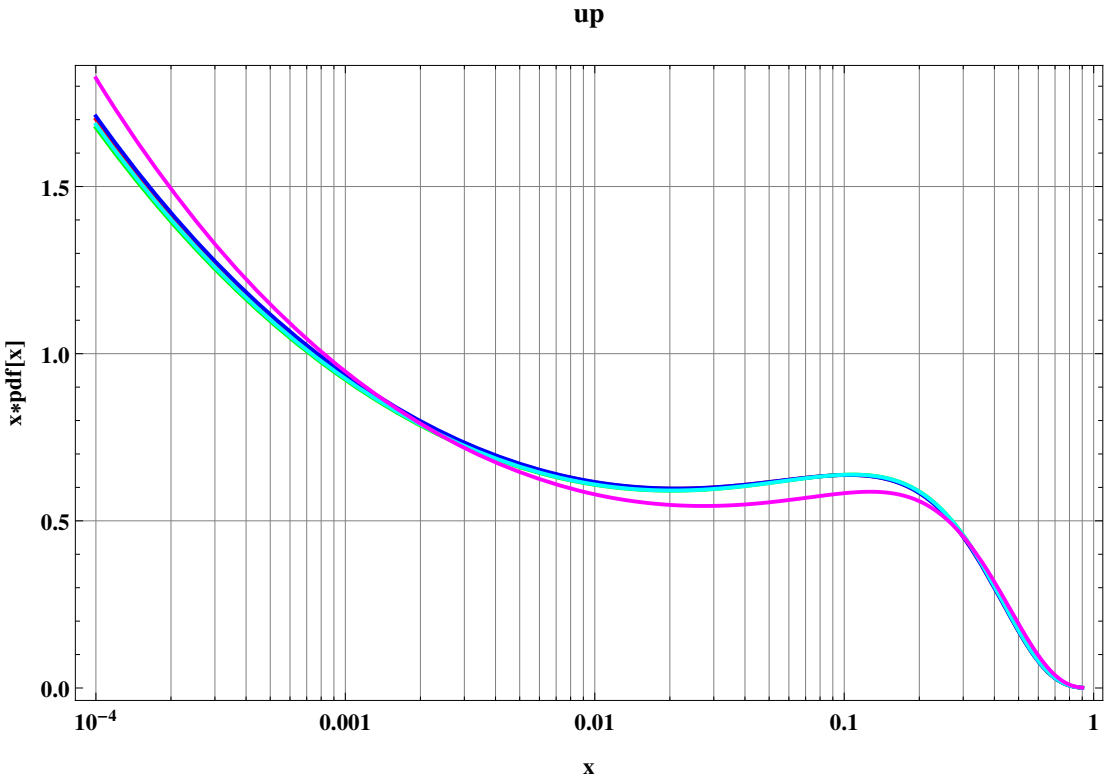
ubar

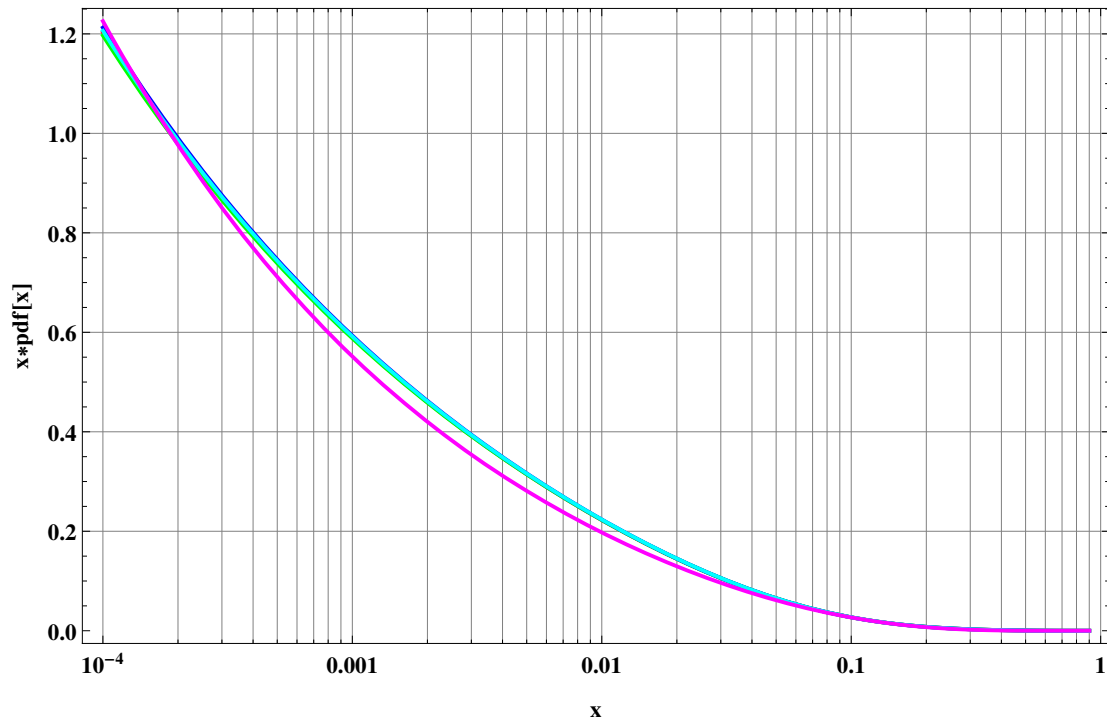
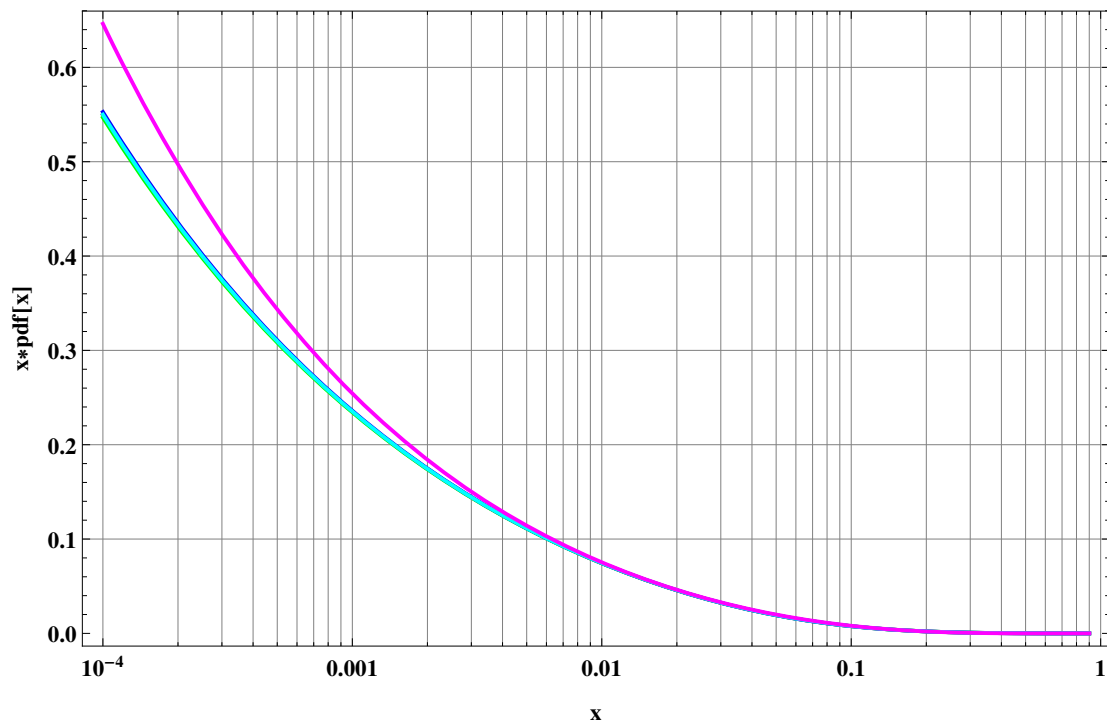


dbar







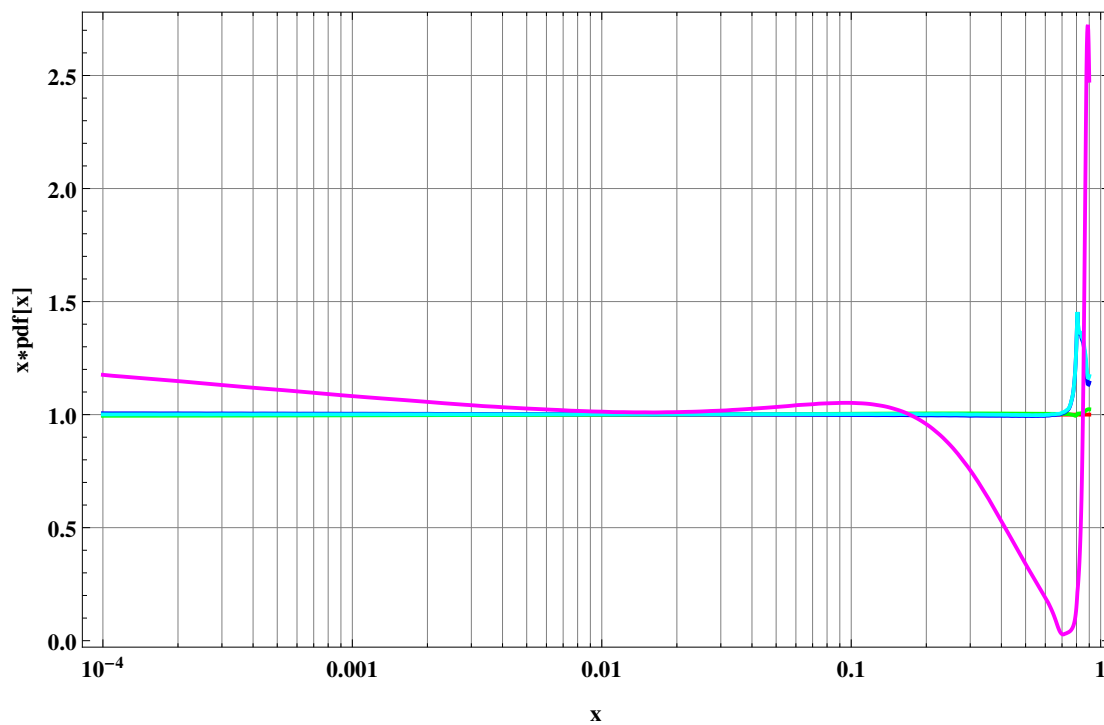
charm**bottom**

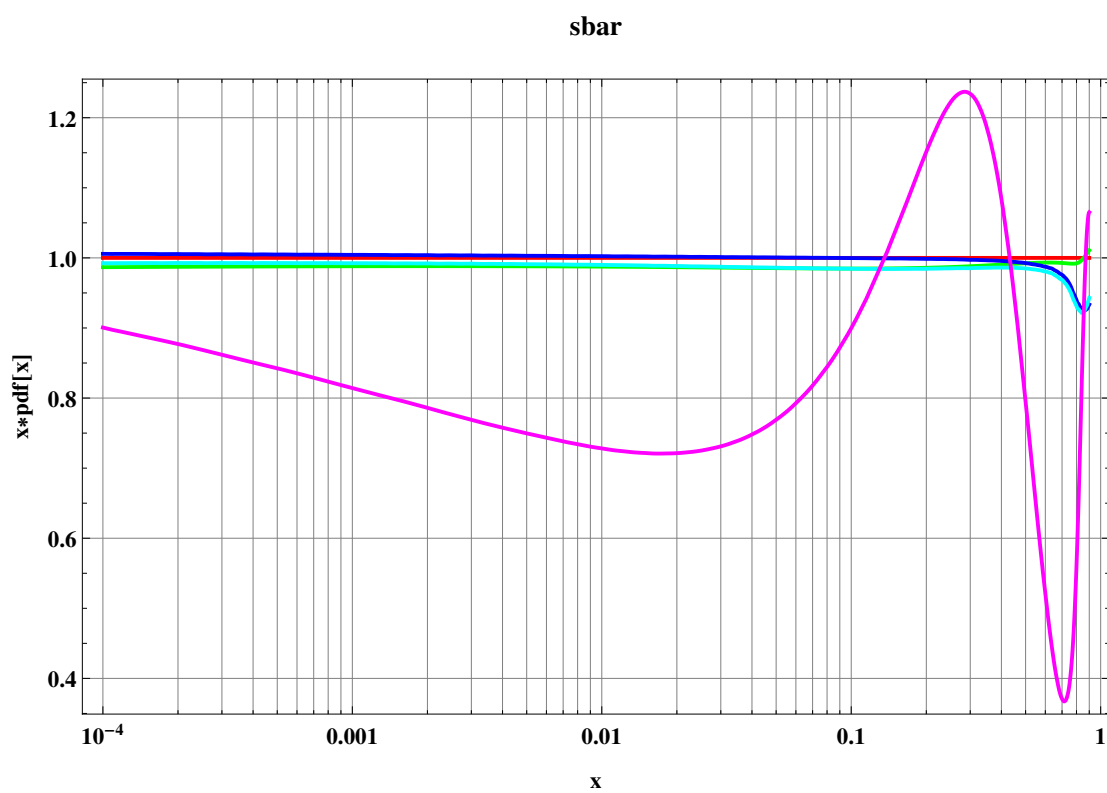
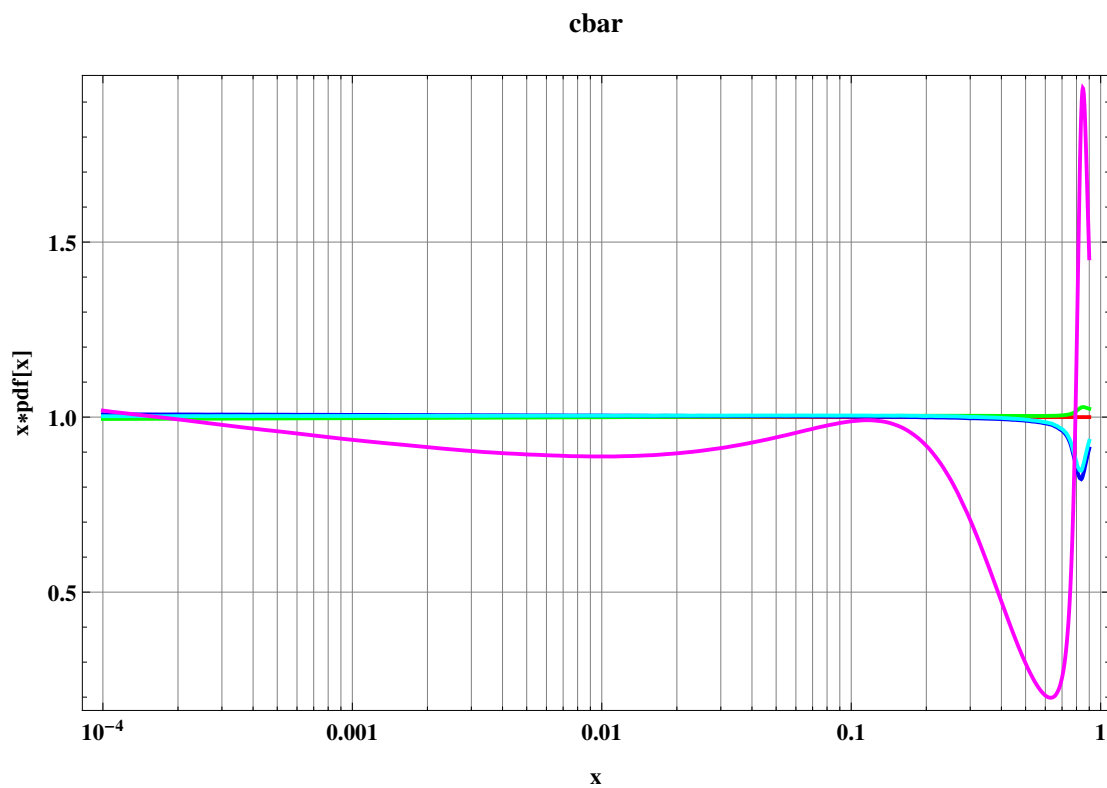
```

For[i = -5, i ≤ 5, i++,
  LogLinearPlot[Table[pdf[iset, i, x, q0] / pdf[1, i, x, q0], {iset, 1, 5, 1}] //
    Evaluate, {x, 10.^-4, 0.9},
    PlotStyle → colors2,
    PlotLabel → pdfFlavorLHA[i],
    FrameLabel → {"x", "x*pdf[x]"},
    ImageSize → Large,
    PlotRange → All,
    Frame → True,
    BaseStyle → {FontWeight → "Bold", FontSize → 12},
    GridLines → Automatic
  ] // Print
]

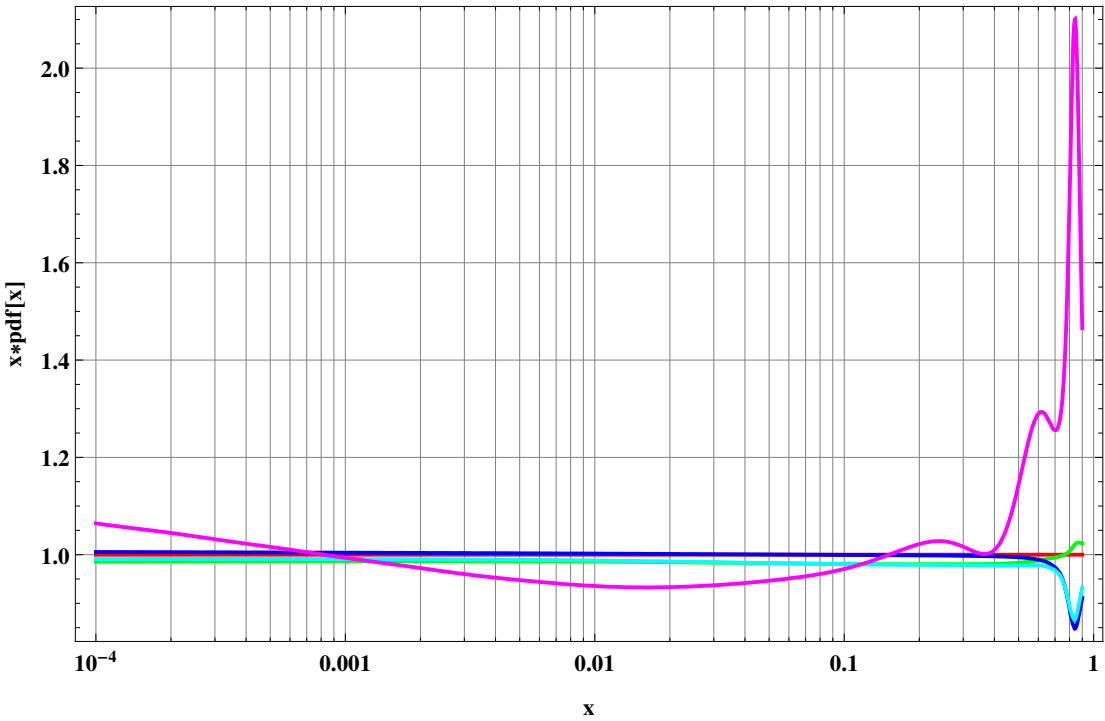
```

bbar

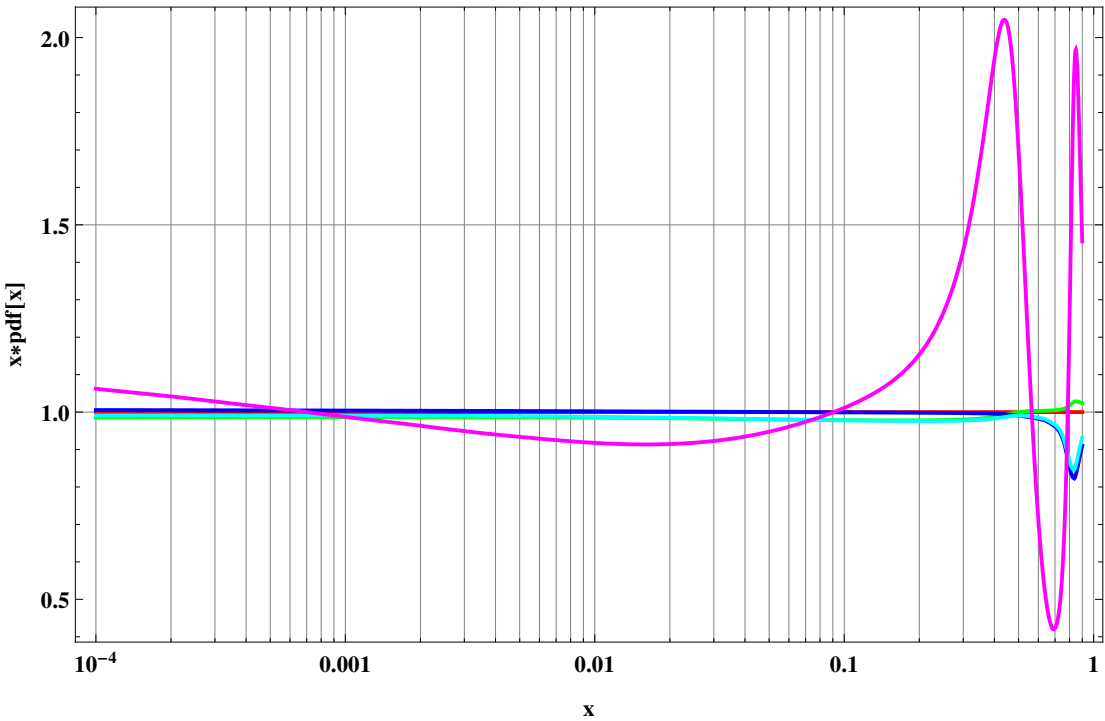


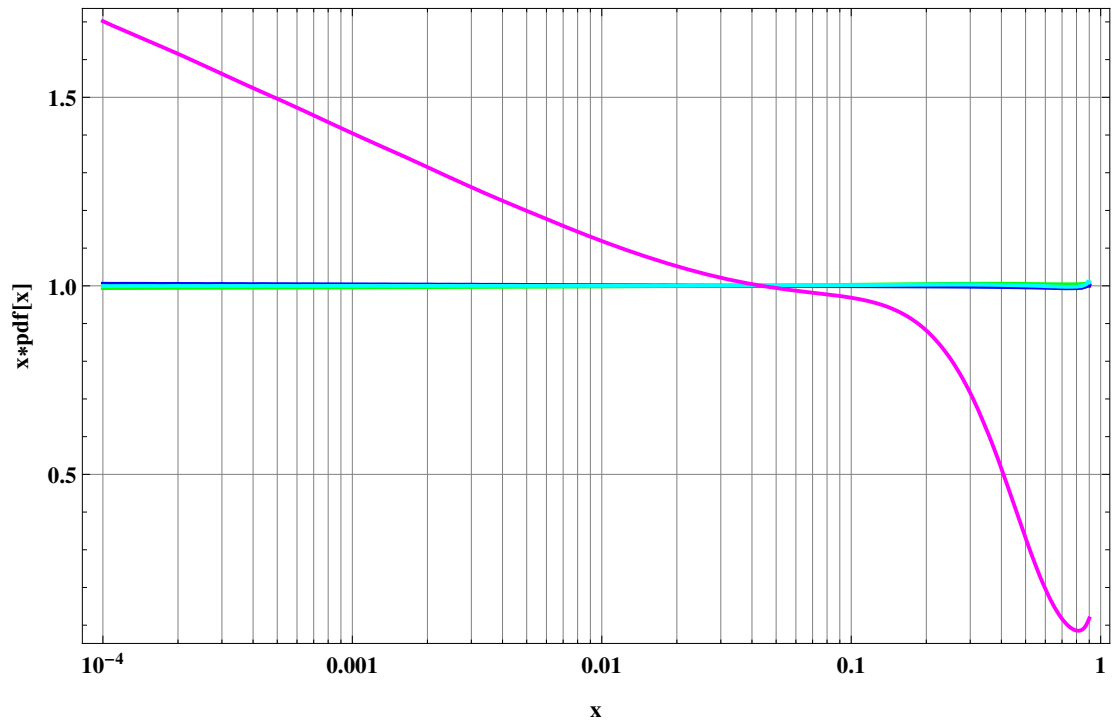
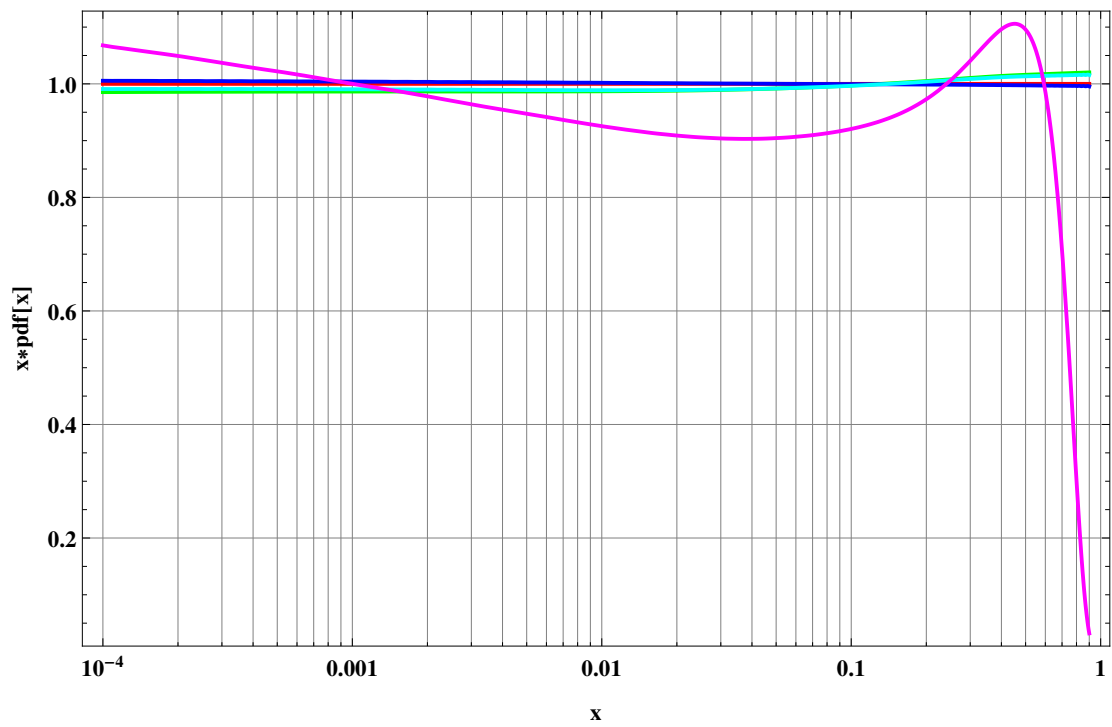


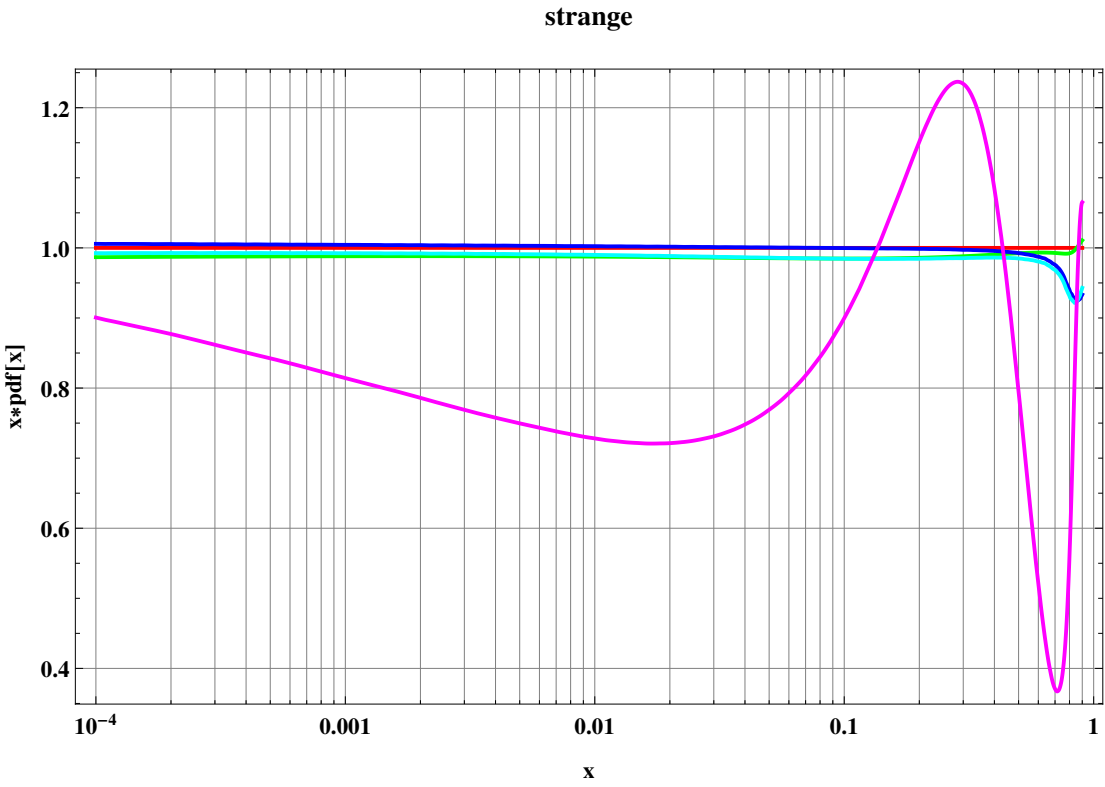
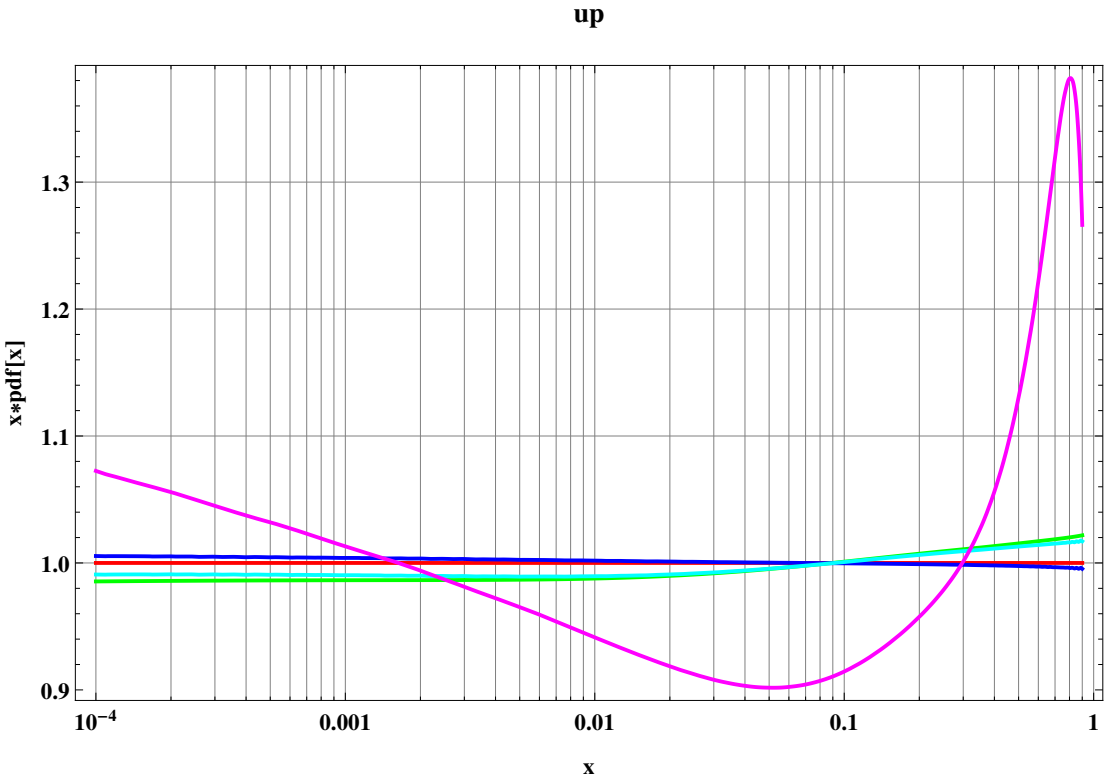
ubar

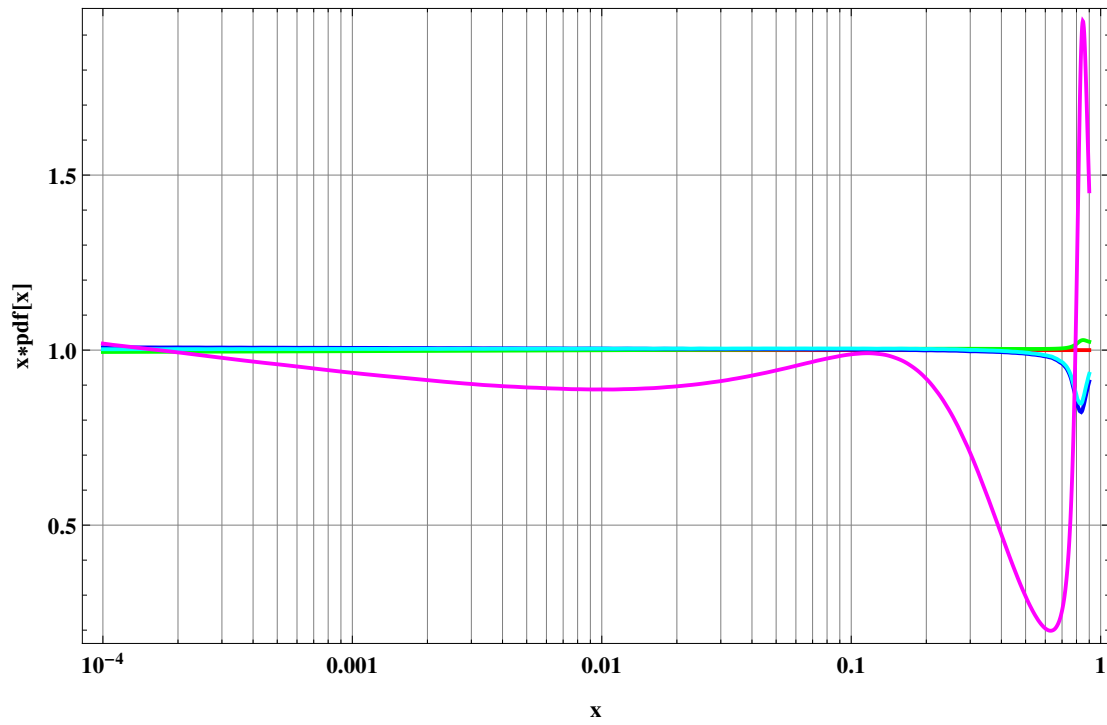


dbar



gluon**down**



charm**bottom**