

# QCD carpentry: 1D structure of the nucleon

---

**Nobuo Sato**  
ODU/JLab

CFNS summer school  
Stony Brook, 2019



# Outline

- **Part I:** Basics of DIS
- **Part II:** Elementary treatment of DIS Factorization
- **Part III:** Solving RGE - coding
- **Part IV:** DIS phenomenology - coding

# Outline

- **Part I:** Basics of DIS
- **Part II:** Elementary treatment of DIS Factorization
- **Part III:** Solving RGE - coding
- **Part IV:** DIS phenomenology - coding

# References

- Collins “Foundations of perturbative QCD”
- Moffat, Melnitchouk, Rogers, NS (PRD95, 2017)
- Vogt (hep-ph/0408244)

# Part IV: Phenomenology

## - coding part

- Fitting PDFs and  $\Delta$ PDFs to global DIS data

# QCD carpentry: coding up a PDF fitter

- reader.py (from the repo)
- residuals.py (from the repo)
- tools.py (from the repo)
- analysis.py (you have to code it up)

## reader.py

```
#!/usr/bin/env python
import sys,os
import numpy as np
import pandas as pd
import params as par
from tools import isnumeric
conf={}

def get_X(tab): ....
def get_W2(tab): ...
def get_idx(tab): ...
def apply_cuts(tab,reaction):...
def modify_table(tab,reaction):...
def load_data_sets(reaction,verb=True):...
```

## reader.py

```
if __name__ == "__main__":
```

```
    conf['datasets']={}
```

```
    #--inclusive unpolarized DIS
```

```
    conf['datasets']['idis']={}
```

```
    conf['datasets']['idis']['xlsx']={}
```

```
    conf['datasets']['idis']['xlsx'][10010]='idis/expdata/10010.xlsx' # proton / F2 /
```

```
    conf['datasets']['idis']['xlsx'][10016]='idis/expdata/10016.xlsx' # proton / F2 /
```

```
    conf['datasets']['idis']['xlsx'][10020]='idis/expdata/10020.xlsx' # proton / F2 /
```

```
    conf['datasets']['idis']['xlsx'][10011]='idis/expdata/10011.xlsx' # deuteron / F2 /
```

```
    conf['datasets']['idis']['xlsx'][10017]='idis/expdata/10017.xlsx' # deuteron / F2 /
```

```
    conf['datasets']['idis']['xlsx'][10021]='idis/expdata/10021.xlsx' # d/p / F2d/F2p /
```

```
    conf['datasets']['idis']['filters']=[]
```

```
    conf['datasets']['idis']['filters'].append("Q2>1.0")
```

```
    conf['datasets']['idis']['filters'].append("W2>4.0")
```

```
    TAB=load_data_sets('idis')
```

## Next steps at command line

```
chmod +x reader.py
```

```
./reader.py
```

# residuals.py

```
#!/usr/bin/env python
import sys
from numpy.random import choice, randn
import numpy as np
import copy
import reader
from reader import conf
import pdf,ppdf,idis

def percent_to_absolute(tabs):...

def add_columns(tabs):...

def get_alpha(tabs):...

def setup_tabs(tabs):...

...
```



## residuals.py

```
def _get_residuals(tabs):
    for idx in tabs:
        npts=len(tabs[idx]['value'])
        for i in range(npts):
            tar=tabs[idx]['target'][i]
            Q2=tabs[idx]['Q2'][i]
            x=tabs[idx]['X'][i]

            if tabs[idx]['obs'][i]=='F2':
                thy=idis.get_F2(x,Q2,tar)
            elif tabs[idx]['obs'][i]=='F2d/F2p':
                thy=idis.get_F2(x,Q2,'d')/idis.get_F2(x,Q2,'p')
            elif tabs[idx]['obs'][i]=='A1':
                thy=idis.get_A1(x,Q2,tar)

            tabs[idx]['thy'][i]=thy
            exp=tabs[idx]['value'][i]
            alpha=tabs[idx]['alpha'][i]
            tabs[idx]['residuals'][i]=(exp-thy)/alpha

    return tabs
```

## residuals.py

```
def set_new_params(par,dist):
    if dist=='pdf':
        pdf.set_params(par)
        pdf.set_sumrules()
        pdf.set_moms()
    if dist=='ppdf':
        ppdf.set_params(par)
        ppdf.set_sumrules()
        ppdf.set_moms()

def get_residuals(par,dist,tabs,verb=False):
    set_new_params(par,dist)
    res=[]
    _get_residuals(tabs)
    for idx in tabs:
        res=np.append(res,tabs[idx]['residuals'])
    print np.sum(res**2),res.size,par
    return res
```

## residuals.py

```
if __name__ == "__main__":  
  
    pdf.set_sumrules()  
    pdf.set_moms()  
    ppdf.set_sumrules()  
    ppdf.set_moms()
```

## residuals.py

```
...
```

```
conf['datasets']={}
```

```
##--inclusive unpolarized DIS
```

```
conf['datasets']['idis']={}
```

```
conf['datasets']['idis']['xlsx']={}
```

```
conf['datasets']['idis']['xlsx'][10010]='idis/expdata/10010.xlsx' # proton / F2 / J
```

```
conf['datasets']['idis']['xlsx'][10016]='idis/expdata/10016.xlsx' # proton / F2 / J
```

```
conf['datasets']['idis']['xlsx'][10020]='idis/expdata/10020.xlsx' # proton / F2 / J
```

```
conf['datasets']['idis']['xlsx'][10011]='idis/expdata/10011.xlsx' # deuteron / F2 / J
```

```
conf['datasets']['idis']['xlsx'][10017]='idis/expdata/10017.xlsx' # deuteron / F2 / J
```

```
conf['datasets']['idis']['xlsx'][10021]='idis/expdata/10021.xlsx' # d/p / F2d/F2p / J
```

```
conf['datasets']['idis']['filters']=[]
```

```
conf['datasets']['idis']['filters'].append("Q2>1.0")
```

```
conf['datasets']['idis']['filters'].append("W2>4.0")
```

```
tabs=reader.load_data_sets('idis')
```

```
tabs=setup_tabs(tabs)
```

```
par=pdf.get_params()
```

```
print get_residuals(par, 'pdf', tabs)
```

Next steps at command line

```
chmod +x residuals.py
```

```
./residuals.py
```

## analysis.py

```
#!/usr/bin/env python
import sys
from numpy.random import choice, randn
import numpy as np
import copy
import reader
from reader import conf
import pdf,ppdf,idis
import residuals
from scipy.optimize import leastsq
from scipy.optimize import least_squares
import idis
from tools import load,save
import pylab as py
...
```

## analysis.py

```
def fit_upol_dis():
```

```
    #--inclusive unpolarized DIS
```

```
    conf['datasets']={}
```

```
    conf['datasets']['idis']={}
```

```
    conf['datasets']['idis']['xlsx']={}
```

```
    conf['datasets']['idis']['xlsx'][10010]='idis/expdata/10010.xlsx' # proton / F2 /
```

```
    conf['datasets']['idis']['xlsx'][10016]='idis/expdata/10016.xlsx' # proton / F2 /
```

```
    conf['datasets']['idis']['xlsx'][10020]='idis/expdata/10020.xlsx' # proton / F2 /
```

```
    conf['datasets']['idis']['xlsx'][10011]='idis/expdata/10011.xlsx' # deuteron / F2 /
```

```
    conf['datasets']['idis']['xlsx'][10017]='idis/expdata/10017.xlsx' # deuteron / F2 /
```

```
    conf['datasets']['idis']['xlsx'][10021]='idis/expdata/10021.xlsx' # d/p / F2d/F2p /
```

```
    conf['datasets']['idis']['filters']=[]
```

```
    conf['datasets']['idis']['filters'].append("Q2>1.0")
```

```
    conf['datasets']['idis']['filters'].append("W2>4.0")
```

```
    tabs=reader.load_data_sets('idis')
```

```
    tabs=residuals.setup_tabs(tabs)
```

```
    ...
```

## analysis.py

```
def fit_upol_dis():  
  
    ....  
  
    ##--setup pdfs  
    pdf.set_sumrules()  
    pdf.set_moms()  
  
    ##--make fit  
    guess=pdf.get_params()  
    #guess=load('pdf.par')  
    bounds=(pdf.parmin, pdf.parmax)  
    res=lambda par: residuals.get_residuals(par, 'pdf', tabs)  
    fit = leastsq (res, guess)  
    save(fit[0], 'pdf.par')  
    #fit = least_squares(res, guess, bounds=bounds, method='trf')#, ftol=1e-4)  
    #save(fit.x, 'pdf.par')
```

## analysis.py

```
def fit_pol_dis():
```

```
    #--inclusive polarized DIS
```

```
    conf['datasets']={}
```

```
    conf['datasets']['pidis']={}
```

```
    conf['datasets']['pidis']['xlsx']={}
```

```
    conf['datasets']['pidis']['xlsx'][10001]='pidis/expdata/10001.xlsx' # 10001 | deuteron |
```

```
    conf['datasets']['pidis']['xlsx'][10033]='pidis/expdata/10033.xlsx' # 10033 | deuteron |
```

```
    conf['datasets']['pidis']['xlsx'][10034]='pidis/expdata/10034.xlsx' # 10034 | deuteron |
```

```
    conf['datasets']['pidis']['xlsx'][10002]='pidis/expdata/10002.xlsx' # 10002 | proton |
```

```
    conf['datasets']['pidis']['xlsx'][10003]='pidis/expdata/10003.xlsx' # 10003 | proton |
```

```
    conf['datasets']['pidis']['xlsx'][10004]='pidis/expdata/10004.xlsx' # 10004 | proton |
```

```
    conf['datasets']['pidis']['xlsx'][10035]='pidis/expdata/10035.xlsx' # 10035 | proton |
```

```
    conf['datasets']['pidis']['xlsx'][10036]='pidis/expdata/10036.xlsx' # 10036 | proton |
```

```
    conf['datasets']['pidis']['filters']=[]
```

```
    conf['datasets']['pidis']['filters'].append("Q2>1.69")
```

```
    conf['datasets']['pidis']['filters'].append("W2>10.0")
```

```
    tabs=reader.load_data_sets('pidis')
```

```
    tabs=residuals.setup_tabs(tabs)
```



## analysis.py

```
def fit_pol_dis():  
    ....  
    #--setup pdfs  
    par=load('pdf.par')  
    pdf.set_params(par)  
    pdf.set_sumrules()  
    pdf.set_moms()  
  
    #--setup ppdfs  
    ppdf.set_sumrules()  
    ppdf.set_moms()  
  
    #--make fit  
    guess=ppdf.get_params()  
    #guess=load('ppdf.par')  
    bounds=(ppdf.parmin,ppdf.parmax)  
    res=lambda par: residuals.get_residuals(par,'ppdf',tabs)  
    fit = least_squares(res, guess,bounds=bounds,method='trf',ftol=1e-4)  
    save(fit.x,'ppdf.par')
```

## analysis.py

```
def fit_pol_dis():
    ....
    #--setup pdfs
    par=load('pdf.par')
    pdf.set_params(par)
    pdf.set_sumrules()
    pdf.set_moms()

    #--setup ppdfs
    ppdf.set_sumrules()
    ppdf.set_moms()

    #--make fit
    guess=ppdf.get_params()
    #guess=load('ppdf.par')
    bounds=(ppdf.parmin,ppdf.parmax)
    res=lambda par: residuals.get_residuals(par,'ppdf',tabs)
    fit = least_squares(res, guess,bounds=bounds,method='trf',ftol=1e-4)
    save(fit.x,'ppdf.par')
```

## analysis.py

```
def plot_pdf():  
    #--setup pdfs  
    par=load('pdf.par')  
    pdf.set_params(par)  
    pdf.set_sumrules()  
    pdf.set_moms()
```

## analysis.py

```
def plot_pdf():  
    ...  
    x=10**np.linspace(-4,-1,100)  
    x=np.append(x,np.linspace(0.101,0.9))  
    Q2=10.0  
    ax=py.subplot(111)  
    ax.plot(x,[pdf.get_xF(_,Q2,'um') for _ in x],label=r'$u_{-}$')  
    ax.plot(x,[pdf.get_xF(_,Q2,'dm') for _ in x],label=r'$d_{-}$')  
    ax.plot(x,[pdf.get_xF(_,Q2,'g') for _ in x],label=r'$g$')  
    ax.plot(x,[pdf.get_xF(_,Q2,'ub') for _ in x],label=r'$\bar{u}$')  
    ax.plot(x,[pdf.get_xF(_,Q2,'db') for _ in x],ls=':',label=r'$\bar{d}$')  
    ax.legend()  
    ax.set_ylim(0,1)  
    ax.set_ylabel(r'$xf(x)$',size=20)  
    ax.set_xlabel(r'$x$',size=20)  
    ax.semilogx()  
    py.tight_layout()  
    py.savefig('xf.pdf')
```

## analysis.py

```
def plot_ppdf():  
    ##--setup ppdfs  
    par=load('ppdf.par')  
    ppdf.set_params(par)  
    ppdf.set_sumrules()  
    ppdf.set_moms()
```

## analysis.py

```
def plot_ppdf():
    ...
    x=10**np.linspace(-4,-1,100)
    x=np.append(x,np.linspace(0.101,0.9))
    Q2=1.2
    ax=py.subplot(111)
    ax.plot(x,[ppdf.get_xF(_,Q2,'up') for _ in x],label=r'$u_+$')
    ax.plot(x,[ppdf.get_xF(_,Q2,'dp') for _ in x],label=r'$d_+$')
    ax.plot(x,[ppdf.get_xF(_,Q2,'g') for _ in x],label=r'$g$')
    ax.plot(x,[ppdf.get_xF(_,Q2,'sp') for _ in x],label=r'$s_+$')
    ax.legend()
    ax.set_ylim(-0.2,0.5)
    ax.set_ylabel(r'$x\Delta f(x)$',size=20)
    ax.set_xlabel(r'$x$',size=20)
    ax.semilogx()
    py.tight_layout()
    py.savefig('xdf.pdf')
```

## analysis.py

```
if __name__ == '__main__':
```

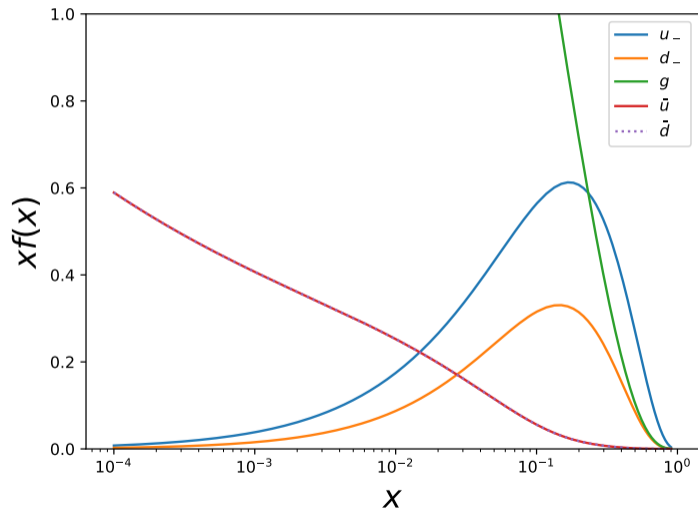
```
    fit_upol_dis()
```

```
    fit_pol_dis()
```

```
    plot_pdf()
```

```
    plot_ppdf()
```

# Unpolarized PDFs fits





## Unpolarized $\Delta$ PDFs fits

