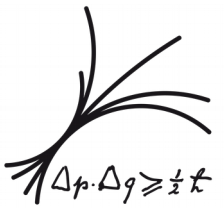
Alpos
an alternative object-oriented fitting package

D. Britzger



Max-Planck-Institut für Physik
(Werner-Heisenberg-Institut)

Alpos

Alpos is a general purpos fitting framework

- Entirely object-oriented

Alpos and its relation to xFitter

- Alpos was built in the surrounding of xFitter, and with benefitting a lot from that experience
- Alpos is also supposed to be a testfield for conceptional (code) developments for xFitter
- Alpos has a fairly different code structure than xFitter, thus facilitating some physics studies, but also complicating other studies

When Alpos started, a number of things had not been very convenient to study in xFitter

- χ^2 evaluation was not flexible, and code was confusing
- Uncertainties with correlation matrices were not fully supported (needed for my code)
- Strong focus on PDFs, but little flexibility for other kind of fits
- Some functionality was difficult to trace back in the code, and understand in greater detail (black box)

Alpos design principles

Design principles

- The code should be strictly object oriented
  → everything should be an instance of a certain object

- Developments by one person should not interfere with studies of others
  → Minimizer, predictions, parameterisations, etc…
  all should be kept separate: i.e. scoped within certain classes

- Any functionality should be well wrapped and also for newcomers quick to identify (no spaghetti code)
  → The code and workflow should 'transparent'

- Flexible data format; flexible uncertainty definition

- Simple things should be easily possible:
  - plain χ^2 calculation
  - simple to define different χ^2 definitions
  - χ^2 for part of the data → possibility to detect 'hot' areas, etc...

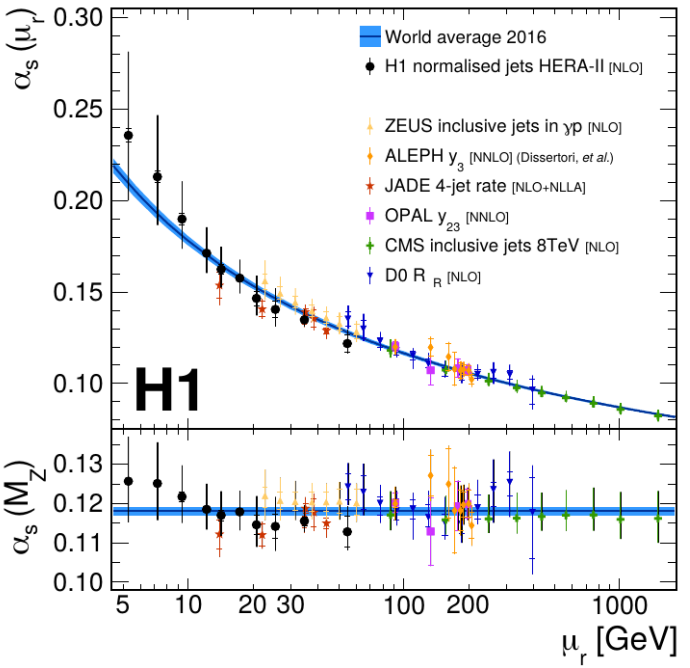
Relevant publications I

H1 low-Q2 jets Eur.Phys.J.C77 (2017) 4

- α_s fit with new H1 ep → jets data
- inclusive jet, dijet and trijet cross sections
- also normalised cross sections: $\sigma_{jet}/\sigma_{NC\text{-}DIS}$
- χ^2 calculation for aNNLO and full NNLO predictions

Tricky:

- data has complicated stat. correlations due to unfolding
  also: inclusive jet, dijet & trijet observables are stat. correlated
  and those correlations are taken into account
- Large amount of theoretical uncertainties had to be evaluated



$$\alpha_s(M_Z) = 0.1172\,(4)_{\text{exp}}\,(3)_{\text{PDF}}\,(7)_{\text{PDF}(\alpha_s)}\,(11)_{\text{PDFset}}\,(6)_{\text{had}}\,(^{+51}_{-43})_{\text{scale}}$$

	n_{dof}	Value of χ^2/n_{dof}					
		NLO	aNNLO	NNLO	NLO	aNNLO	NNLO
		Absolute jet cross sections			Normalised jet cross sections		
Inclusive jet at low-Q^2	48	1.7	2.1	0.8	1.9	1.6	1.3
Inclusive jet at low- and high-Q^2	78	1.7	2.0	1.3	1.9	2.2	2.2
Dijet at low-Q^2	48	1.4	1.9	0.6	1.6	1.7	1.0
Trijet at low-Q^2	32	0.6			0.6		

Relevant publications II

α_s *from global inclusive jets*

Eur.Phys.J. C79 (2019) 68

- αs from 'global' inclusive jet cross section data:
  One data set from:
  H1, ZEUS, STAR, CDF, D0, ATLAS, CMS

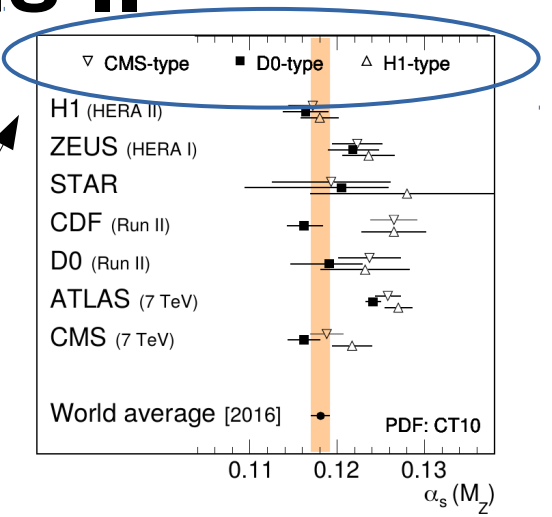
Technical advancement:

- α_s extrations from D0, H1, CMS have been repeated
  → their fitting methods differ in many aspects:

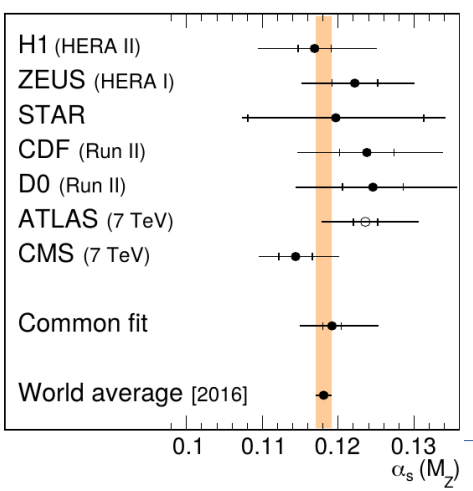
> The three extraction methods differ in the following aspects:
> – the definition of the χ^2 function to quantify the agreement between theory and data,
> – the uncertainties considered in the χ^2 function,
> – the strategy to determine the central result for $\alpha_s(M_Z)$,
> – the propagation of the uncertainties to the value of $\alpha_s(M_Z)$,
> – the choice of PDF sets,
> – the consideration of the $\alpha_s(M_Z)$ dependence of the PDFs, and
> – the treatment of further theoretical uncertainties.

these 3 fitting procedures have been successfully re-implemented in Alpos (including uncertianties)
→ This demonstrates great flexibility of the Alpos framework

- One 'common' type fit strategy has been developed



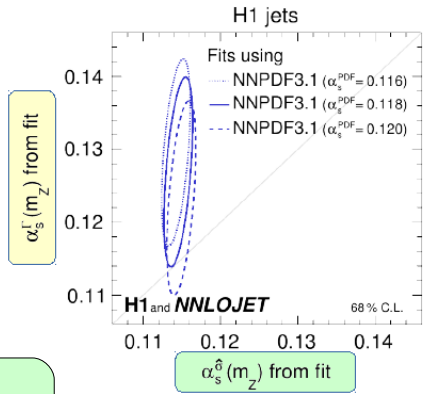
Fit method	CMS-type		D0-type		H1-type	
PDF set	MSTW2008	CT10	MSTW2008	CT10	MSTW2008	CT10
Data	$\alpha_s(M_Z)$ values with experimental uncertainties					
H1	0.1172 (28)	0.1172 (28)	0.1161 (27)	0.1164 (26)	0.1174 (22)	0.1180 (22)
ZEUS	0.1213 (28)	0.1223 (29)	$0.1210\,(^{+28}_{-29})$	$0.1218\,(^{+30}_{-29})$	0.1231 (30)	0.1236 (30)
STAR	—	0.1193 (68)	—	$0.1205\,(^{+54}_{-111})$	0.1159 (116)	0.1280 (111)
CDF	0.1217 (17)	0.1265 (27)	$0.1202\,(^{+10}_{-27})$	$0.1162\,(^{+22}_{-20})$	0.1217 (35)	0.1265 (37)
D0 (22 pts., NLO)	0.1226 (32)	0.1237 (36)	0.1203 $(^{+40}_{-42})$	$0.1191\,(^{+38}_{-45})$	0.1219 (50)	0.1232 (51)
ATLAS	0.1220 (9)	0.1258 (15)	$0.1204\,(^{+14}_{-5})$	0.1241 (9)	0.1206 (15)	0.1270 (16)
CMS	0.1162 (14)	0.1188 (19)	0.1158 (12)	0.1162 (19)	0.1140 (21)	0.1217 (23)



Relevant publications III

α_s from H1 jet data in NNLO Eur.Phys.J.C77 (2017), 791

First use of NNLO grids for α_s determination and/or PDF fit



Technical achievements: Two distinct fits

PDFs $\frac{\partial f}{\partial \alpha_s} = \frac{\mathcal{P} \otimes f}{\beta}$

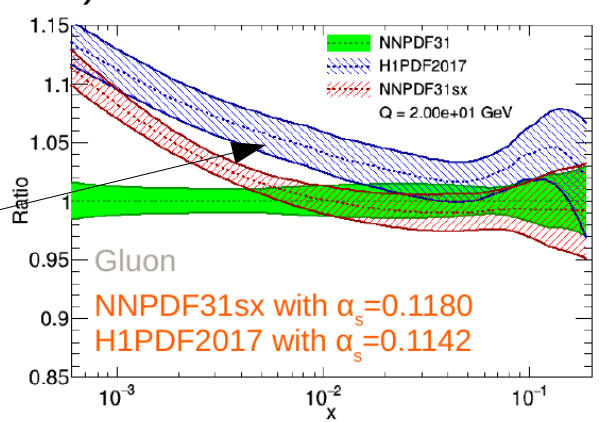
Hard ME's $\hat{\sigma}_{i,k}^{(n)} = \alpha_s^n(\mu_R)\tilde{\sigma}_{i,k}^{(n)}(x, \mu_R, \mu_F)$

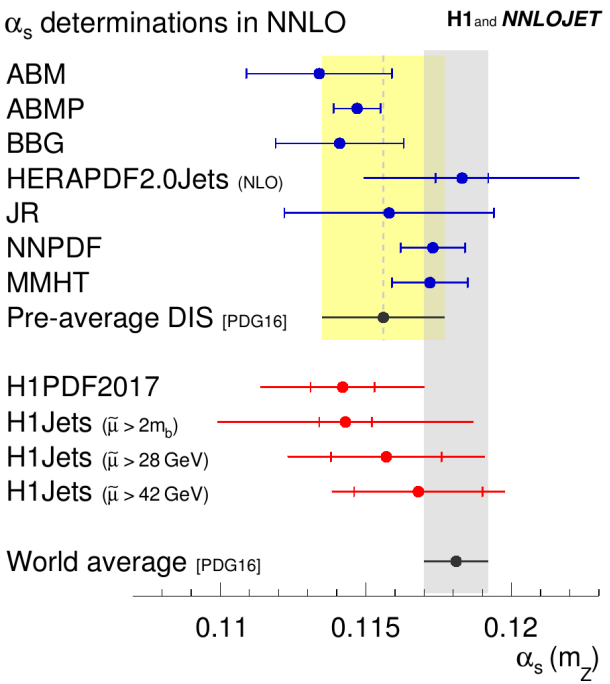
1) fit of α_s to ep jet cross sections

$$\sigma_i = \sum_{n=1}^{\infty} \sum_{k=g,q,\overline{q}} \int dx f_k(x, \mu_F) \hat{\sigma}_{i,k}^{(n)}(x, \mu_R, \mu_F) \cdot c_{\text{had}}$$

both α_s dependencies are accounted for, by defining μ_0=20GeV and performing DGLAP (using QCDNUM, APFEL or APFEL++)

2) Fit to NC & CC DIS from H1 + norm. jet cross sections → H1PDF2017



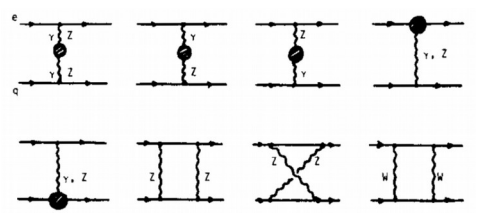


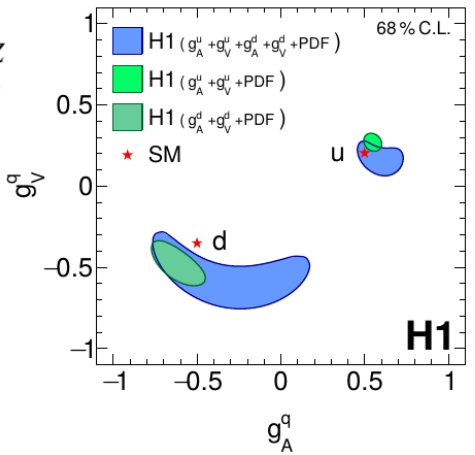
Electroweak parameters from HERA incl. DIS data (H1)

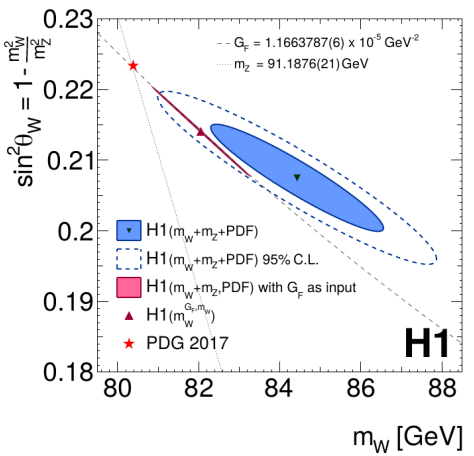
- PDF+EW fit to all H1 incl. NC & CC DIS data (using QCDNUM) Eur.Phys.J. C78 (2018) 777
- 1-loop EW corrections obtained from EPRC

$$\tilde{F}_2^{\pm} = F_2 - (g_V^e \pm P_e g_A^e)\varkappa_Z F_2^{\gamma Z} + \left[(g_V^e g_V^e + g_A^e g_A^e) \pm 2P_e g_V^e g_A^e\right]\varkappa_Z^2 F_2^Z$$

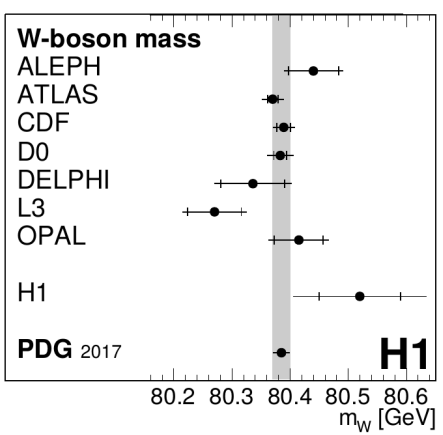
$$\left[F_2, F_2^{\gamma Z}, F_2^Z\right] = x\sum_q \left[Q_q^2, 2Q_q g_V^q, g_V^q g_V^q + g_A^q g_A^q\right]\{q + \bar{q}\}$$

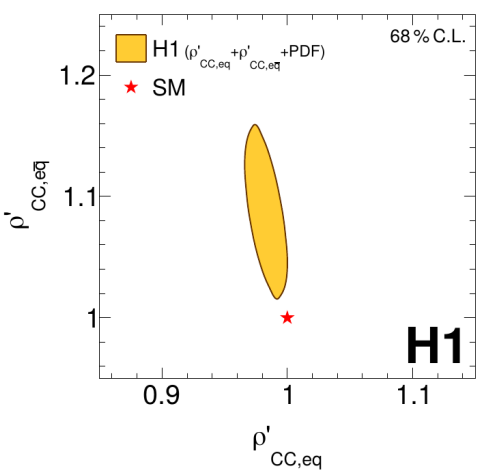


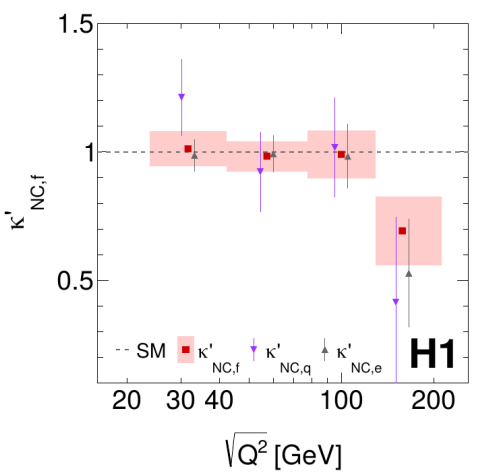




	Data set	Q^2-range [GeV^2]	$\sqrt{s}$ [GeV]	$\mathcal{L}$ [pb^{-1}]	No. of data points	Polarisation [%]	Ref.
1	e^+ combined low-Q^2	(0.5) 8.5 – 150	301,319	20, 22, 97.6	94 (262)	–	[56]
2	e^+ combined low-E_p	(1.5) 8.5 – 90	225,252	12.2, 5.9	132 (136)	–	[56]
3	e^+ NC 94–97	150 – 30 000	301	35.6	130	–	[32]
4	e^+ CC 94–97	300 – 15 000	301	35.6	25	–	[32]
5	e^- NC 98–99	150 – 30 000	319	16.4	126	–	[33]
6	e^- CC 98–99	300 – 15 000	319	16.4	28	–	[33]
7	e^- NC 98–99 high-y	100 – 800	319	16.4	13	–	[57]
8	e^+ NC 99–00	150 – 30 000	319	65.2	147	–	[57]
9	e^+ CC 99–00	300 – 15 000	319	65.2	28	–	[57]
10	e^+ NC L HERA-II	120 – 30 000	319	80.7	136	−37.0 ± 1.0	[58, 59]
11	e^+ CC L HERA-II	300 – 15 000	319	80.7	28	−37.0 ± 1.0	[58, 59]
12	e^+ NC R HERA-II	120 – 30 000	319	101.3	138	+32.5 ± 0.7	[58, 59]
13	e^+ CC R HERA-II	300 – 15 000	319	101.3	29	+32.5 ± 0.7	[58, 59]
14	e^- NC L HERA-II	120 – 50 000	319	104.4	139	−25.8 ± 0.7	[58, 59]
15	e^- CC L HERA-II	300 – 30 000	319	104.4	29	−25.8 ± 0.7	[58, 59]
16	e^- NC R HERA-II	120 – 30 000	319	47.3	138	+36.0 ± 0.7	[58, 59]
17	e^- CC R HERA-II	300 – 15 000	319	47.3	28	+36.0 ± 0.7	[58, 59]
18	e^+ NC HERA-II high-y	60 – 800	319	182.0	11	–	[58, 59]
19	e^- NC HERA-II high-y	60 – 800	319	151.7	11	–	[58, 59]



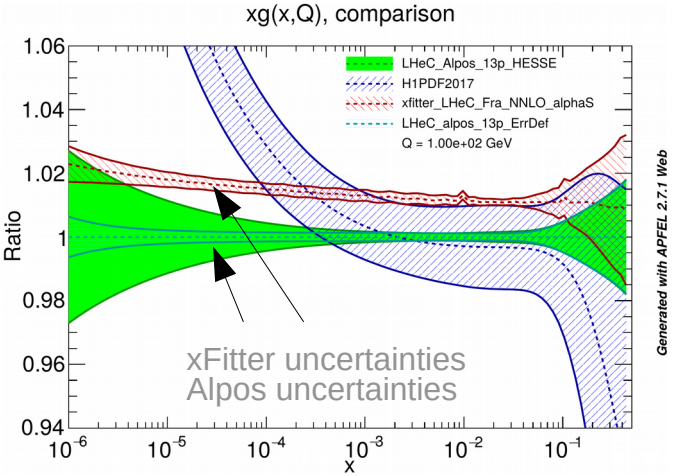


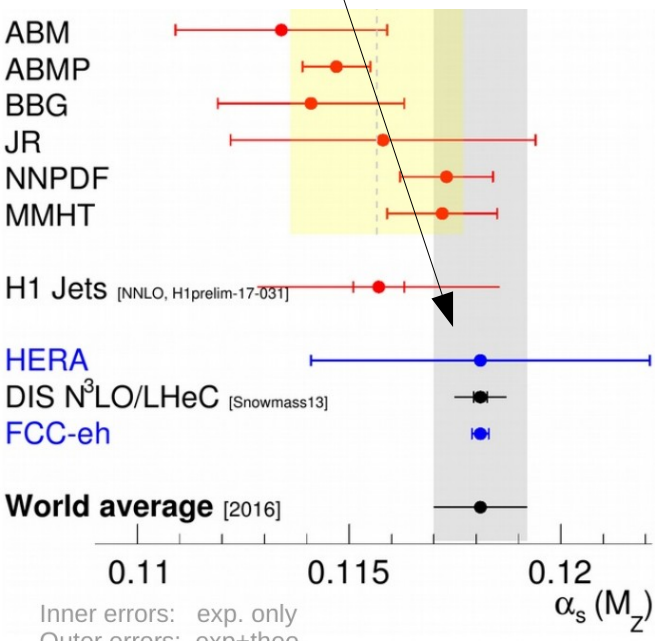


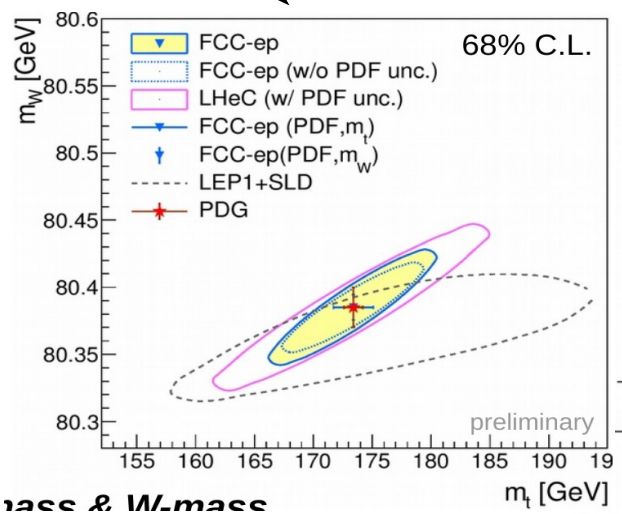
FCC-eh/LHeC future perspectives

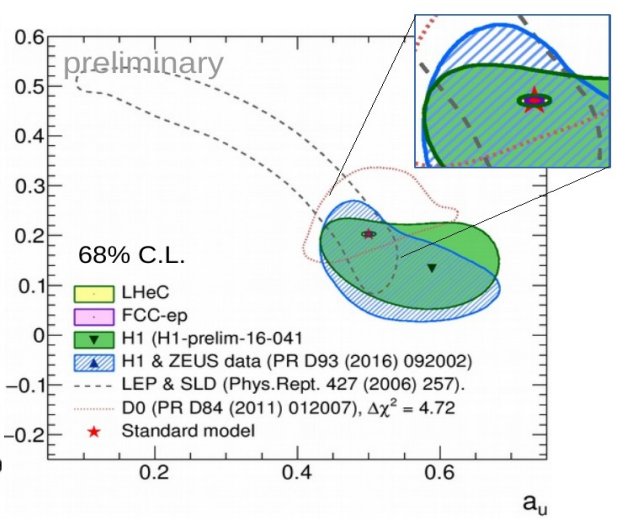
FCC-eh and LHeC perspectives

- PDFs (benchmarking xFitter results [Fra Giuli])
- Electroweak physics with incl. DIS
  e.g. m_{top} through 1-loop EW corrections
- α_s prospects





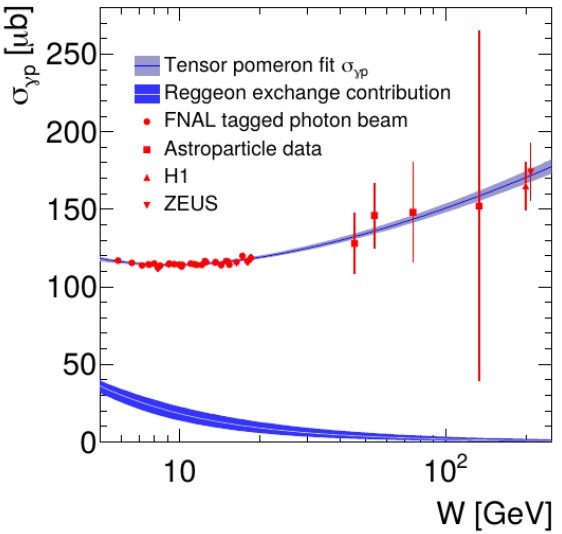




Relevant publications IV

Two tensor pomeron fit arXiv:1901.08524

- Fit of two-tensor pomeron model to inclusive NC DIS and PHP data
  → See Sasha's talk on tuesday

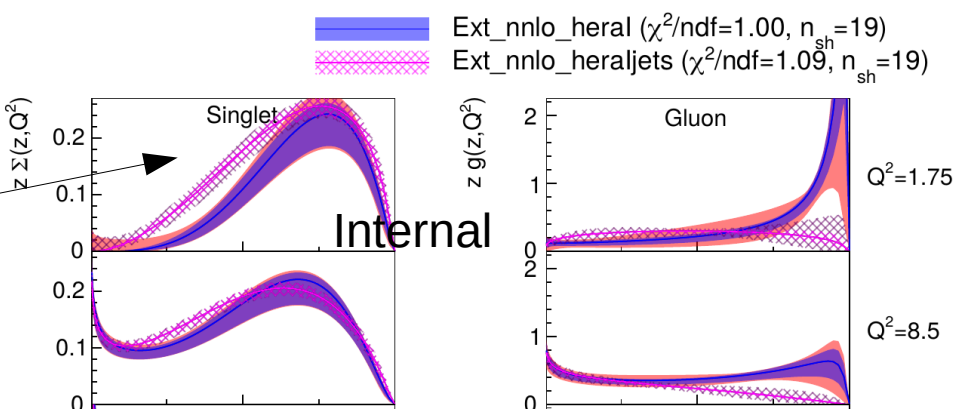


α_s fit from dijet data in diffractive DIS in NNLO Eur.Phys.J. C78 (2018) 538

- χ^2/ndf=13/14
- $\Delta\alpha_s$ = 4%

Ongoing:

- α_s from HERA inclusive jets (DIS19)
- 2D limits on contact interactions with HERA inclusive DIS data (~2020)
- Belle time-dependent CP violation analysis
  → Vladimir's talk
- Diffractive PDFs with H1 data and dijet cross sections in NNLO (DIS19)
  APFEL, APFEL++, QCDNUM, GRV-pion, Owen's pion, H1DPDF2006, FONLL-C, FFNS, ZMVFNS, NNLO diffractive dijets w/ fastNLO+NNLOJET, arbitrary parameterisations, χ^2 definitions, 3D + 4D diff. data, ...



Alpos: code structure

In the following, I will rather focus on technical details, and underlying programming concepts

- This may be pretty technical...
- Probably, some ideas advance upcoming xfitter developments.

Design considerations

Flexible to use

- α_s fits
- Electroweak fits
- Simple χ^2 studies (without fits)
- PDF fits

It should be VERY simple to implement new theory prediction

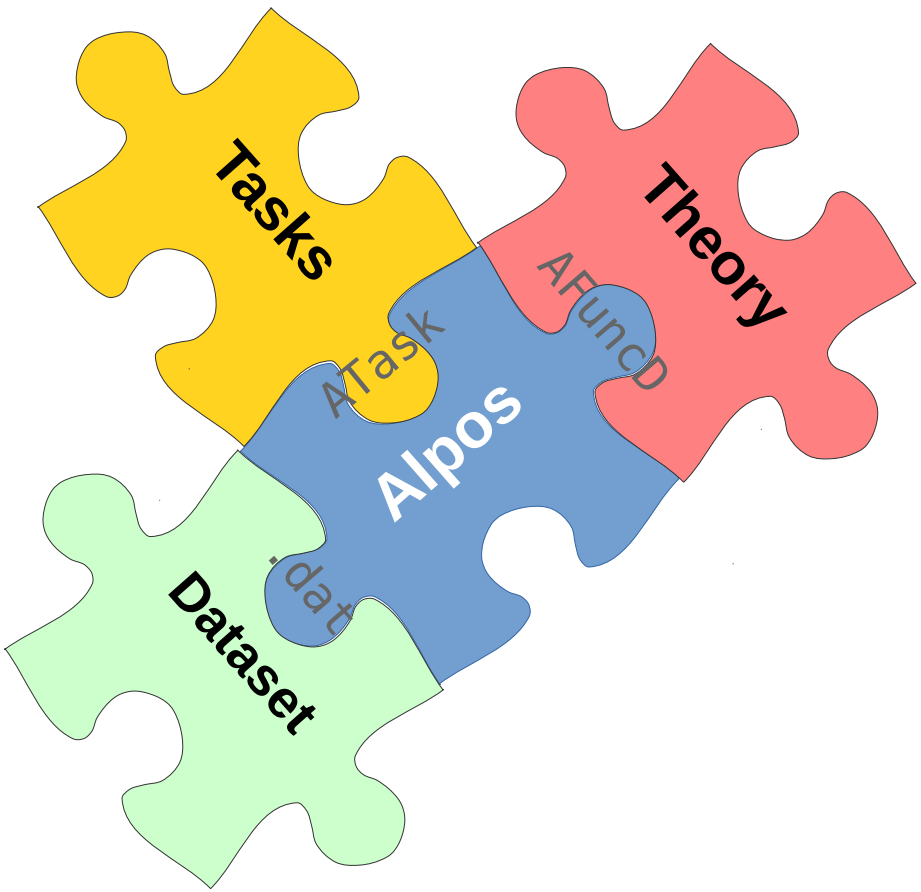
It should be VERY simple to implement a new minimizer (or similar tasks)

Support for flexible data formats

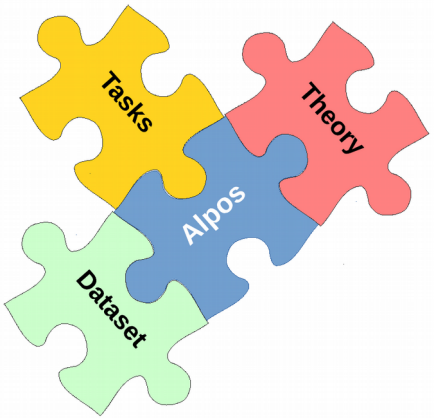
- Many uncertainties, with different kind of uncertainty definitions/correlations

Funtionalities for detailed data-to-theory studies

- Detection hot area's → 'subsets'
- partial χ^2, for different χ^2 definitions
- Transparent workflow



Technicalities I



Alpos

- Program execution

```
$ alpos tutorial/6.herapdf10.str
```

- Program workflow

```
#include "alpos/Alpos.h"
int main(int argc, char** argv) {
   string steerfile = argv[1];
   Alpos alpos(steerfile,argc-1,&argv[1]);
   alpos.DoTasks();
   return 0;
}
```

Hint: arbitrary additional information can be passed through the command line using equivalent syntax as steering file: Values are superseeding keys from steering

Initialisation

- Parse steering file (*.str), and Read data files (*.dat) from steering
- Initialize theory 'functions' as specified in steering

Run list of 'tasks', as specified in steering

- Run subsequently 'tasks'

Note: code examples are sometimes shortened!

Technicalities II

Essentially, the Alpos package consists only of

few (~two) very simple abstract classes:

- For performing calculations, interfacing predictions, and organising fit parameters

```
template<typename T> class AParm          : public AParmBase<T>
template<typename T> class AParmFuncBase : public AParmBase<T>
```

- For the tasks:

```
class ATask       : public AlposObject
class ATaskResult : public AlposObject
```

- A simple singleton class '`ATheoryHandler`'

  keep track of all parameters (all are fittable), all relations between parameters and functions, etc… [next slides]

- Three factories

```
namespace AFactory {
   AFuncD*     FunctionFactory(const std::string& functype,const std::string& funcname); //! init a new function
   ATask*      TaskFactory    (const std::string& tname, const std::string& ttype);      //!< instantiat a new task
   AChisqBase* ChisqFactory   (const std::string& chisq, const std::vector<std::string>& par, AData* data, AFuncD* theo);
}
```

The 'TheoryHandler' class

The TheoryHandler

- a singleton class

```
class TheoryHandler : public AlposObject {
private:
   TheoryHandler();

public:
   static TheoryHandler* Handler();
   bool InitTheory(const std::string& steerfile = std::string());
   /* [...]*/

private:
   static TheoryHandler* instance;
   std::map<std::string, AParmNamed*> fParms;
   std::set<std::string> fAllParmNames;
   std::pair<ASuperData*, ASuperTheory*> fSuperPair; // map of datasets and corresponding predictions
   std::map<std::string, std::pair<AData*, AFuncD*> > fDataTheoryPairs;  // map of datasets and corresponding predictions
   std::map<std::string, std::map<std::string, std::pair<ASubsetData*, ASubsetFunction*> > > fSubsetPairs;
};
```

First, it initialises the theory module (i.e. all parameters and functions)
as specified in the steering file

it 'owns' all parameters (and functions) and holds them in a map

There are 1:1 relations between a 'data function' and some 'theory function':

Background: any 'data' can only be specified in the steering with some related 'theory function' to provide its predictions

Everything else here…
`ASuperData`, `ASuperTheory`, `AData`, `ASubsetData`, `ASubsetFunction`, ...,
are just implementations of abstract class
`typedef AParmFuncBase<double> AFuncD;`

Parameters and functions

Parameters
`template<typename T> class AParm`

Functions
`template<typename T> class AParmFuncBase`


```
class AParmNamed : public AlposObject {
public:
   AParmNamed(const std::string& name);
   virtual bool Update() { return true; };
   void SetIsOutdated(AParmNamed* notifier = nullptr, bool IsOutdated = true);
   void NotifyDependencies(AParmNamed* notifier);

private:
   bool fIsUpToDate = false;
   std::map<std::string, AParmNamed*> fDependencies; //!< List of parm's which are going to be notified, if parm is being updated.
```

1) Every '`AParmNamed`' instance knows, which other `AParmNamed` makes use of it
2) If 'something' has changed, it notifies all upstream dependent instances, that also those need to re-redo their calculations

```
template<typename T> class AParmBase : public AParmNamed {
public:
   AParmBase(const std::string& name);
   const std::vector<T>& GetValues() {
      if ( !GetIsUpToDate() )  Update();
      ApplyThFactors();
      return fValue;
   }
   const T& GetValue() { return GetValues()[0]; }
   const T& GetError() { return GetErrors()[0]; }
   /*[...]*/
protected:
   virtual bool Update() = 0;
   std::vector<T> fValue;
   std::map<std::string, vecT> fThFactors;
};
```

1) Every '`AParmBase`' has a vector<T> of values

2) The values are obtained with 'Update()'. Check 'fUpToDate'
   - Redo calculation of that implementation
   - else… return cached value

Inherent caching algorithm !

Example implementation of a 'function': fastNLO

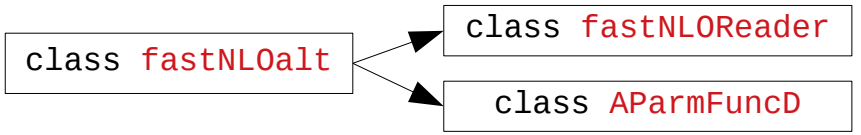
```
// ____________________________________________________________________
const std::vector<std::string> AfastNLOalt::fRequirements = {
   "Filename",
   "PDF",
   "Alpha_s",
   "ScaleFacMuR","ScaleFacMuF",
   "Units","iOrd",
   "MuRFuncForm","MuFFuncForm"
}; //< List of all AParm's which this function depends on
const std::string AfastNLOalt::fFunctionName = "fastNLOalt"; //< The function's name


// ____________________________________________________________________
bool AfastNLOalt::Init() {
   ReadTable();
   fastNLOReader::SetFilename(PAR_S(Filename)); // do some remaining initialization
   return true;
}

// ____________________________________________________________________
bool AfastNLOalt::Update() {
   UPDATE(PDF);
   UPDATE(Alpha_s);
   /* [...] */
   CalcCrossSection();
   fValue = GetCrossSection();
   return true;
}

// ____________________________________________________________________
double AfastNLOalt::EvolveAlphas(double Q) const { //fastNLO
   return QUICK(Alpha_s, ({Q}) )[0];
}

// ____________________________________________________________________
vector<double> AfastNLOalt::GetXFX(double x, double Q) const { //fastNLO
   return QUICK(PDF, ({x,Q}) );
}
```



Here: almost entire interface of fastNLO in Alpos (little bit simplified)

1) fRequirements:
   These 'parameters' or 'functions need to be specified in the steering as input to fastNLO
   - 'Filename' is the fastNLO filename
   - PDF,Alpha_s are 'functions'
   - ScaleFacMuR is a 'parameter' holding the scale factor
   - […] few more parameters (removed from that example)
2) Init(): once called during startup
3) Update(): is called, once some of the 'input' parameters (fRequirements) has changed, a
4) fastNLO specific interface of PDFs and $\alpha_s(\mu_R)$, accessing the 'Update()' return from the Alpos functions 'PDF' and Alpha_s

Technicalities IV: steering file, parameters

```
# ------------------------------------------------------------#
#  Alpos global settings
# ------------------------------------------------------------#
ErrorSymmetrization             SignImprovedQuadratic
Output                          test/alpos.out.root             # common output file for outputs of all tasks
GlobalVerbosity                 Info                            # Debug, Warning, Error
InitSubsets                     false                           # init subsets of datasets
IgnoreTheoryErrors              true                            # Ignore all theory errors when calculating chi2 values
# ALPOS_DIR                       <specify your directory>      # use env-variable ALPOS_DIR or specify it herewith
MatrixInversionTolerance        1.e-6                           # Tolerance of matrix inversion algorithm

# ------------------------------------------------------------#
#  Specify data-files with corresponding predictions
# ------------------------------------------------------------#
DataTheorySets {{
   AlposName    SteerFile   TheoryFunction    CutsAndParameters # header
   # no data files in that example...                           # ...
}}

Tasks {{
  TaskName            TaskType                                  # header
  pi                  3.1                                       # trival task: set parameter 'pi' to value 3.1
  PrintTheorySet      PrintTheorySet                            # print all the theory parameters
}}

################################################################
#  Specify Alpos theory
################################################################
AlposTheory {{{
   #   Specify functions
   InitFunctions {{                                             # Specify all needed alpos functions (name,type)
     FunctionName            FunctionType                       # one line here! (header of 'table')
     # no function in that example....
   }}

   # specify some parameters...
   e                            2.7182818
   pi                           e
   city                         Minsk

}}} # end of 'AlposTheory' namespace
# ------------------------------------------------------------#
```

global parameters for Alpos

In this example: No data

Two trival tasks:
1) change value of parameter 'pi'
2) print all parameters

Functions
(in this example: no functions)

Parameters:
here: e and pi are set to be equal !

Output: $> alpos minsk.str

Parameter 'pi' is set to 3.1

Parameters 'pi' and 'e' are effectively the same object: thus they have the same value

```
+-------------------------------------------------------------------------------------------------------+
+ ---------------------------- Alpos::DoTasks(). Instantiating new task.
+ ----------------------------        Task type: pi
+ ----------------------------        Task name: 3.1
+-------------------------------------------------------------------------------------------------------+


# INFO.    [AFactory::TaskFactory] Initializing task of type 'pi', as name '3.1'.
# INFO.    [Alpos::DoTasks] Task type was identified to be a parameter.
# INFO.    [Alpos::DoTasks] Now setting parameter 'pi' to new value: '3.1'
# INFO.    [Alpos::DoTasks] Parameter set successfully. Now proceeding to next task.


+-------------------------------------------------------------------------------------------------------+
+ ---------------------------- Alpos::DoTasks(). Instantiating new task.
+ ----------------------------        Task type: PrintTheorySet
+ ----------------------------        Task name: PrintTheorySet
+-------------------------------------------------------------------------------------------------------+


# INFO.    [AFactory::TaskFactory] Initializing task of type 'PrintTheorySet', as name 'PrintTheorySet'.
# INFO.    [Alpos::DoTasks] Task 'PrintTheorySet' instantiated.
# INFO.    [Alpos::DoTasks] Running the task 'PrintTheorySet' of type 'PrintTheorySet'.

 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 +  SuperData                                                          [empty]
 +  SuperTheory                                                       [empty]
 +  city                                                              Minsk
 +  e                                                                 3.1
 +  pi  --> e                                                         3.1
 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


# INFO.    [Alpos::DoTasks] Task 'PrintTheorySet' done successfully. Runtime 00:00:00

---------------------------------------------------------------------------------------------
# INFO.    [Alpos::DoTasks] All tasks done [1/2].

Alpos total runtime 00:00:0.54
```

Technicalities IV: steering file, parameters

```
DataTheorySets {{
   AlposName           SteerFile                                                  TheoryFunction    CutsAndParameters
   H1-InclJet-HERAII   datafiles/h1/1406.4709/H1-HERAII-HighQ2-InclJets.dat     fastNLO           # ptmin>=11 q2min>333
}}

# ------------------------------------------------------------#
#  Specify tasks which should be executed by Alpos
# ------------------------------------------------------------#
Tasks {{
   TaskType            TaskName                              # first line is the 'header'
   PrintTheorySet      PrintTheorySet
   AFitter             MyFit                                 # Tasks with name(=steering namespace below)
   PrintTheorySet      PrintTheorySet
   StatAnalysis        Stat                                  #
}}

# ------------------------------------------------------------#
#  Task parameters
#    Put all parameters in a namespace with the task's name
# ------------------------------------------------------------#
MyFit {{{
   Minimizer                  TMinuit
   PrintLevel                 3
   Tolerance                  0.1
   Strategy                   2
   Chisq                      LogNormal #HERAFitterLogDefault
   PrintResults               true
   PrintCovariance            true

   FitParameters { AlphasMz
                   PDFQ0_HERA.gB  PDFQ0_HERA.gC }

}}}

Stat {{{
   Chisq                      LogNormal
   Chisq2                     LogNormalNuisance #HERAFitterDefault
   DoChisq                    true
   DoPValue                   true
}}}

################################################################
#  Specify Alpos theory (in 'AlposTheory' namespace)
################################################################
AlposTheory {{{
# [...]
```

A list of data sets, and corresponding 'functions' providing their predictions

A list of tasks, that Alpos should work through. Every task type can be executed multiple times for different purposes

Steering values for the tasks…
Here:
'MyFit' is of type 'AFitter'
(see 'Tasks')

Setup of the 'AlposTheory'
(next slide)

AlposTheory in steering file: a simple example

```
DataTheorySets {{
   AlposName          SteerFile                                          TheoryFunction    CutsAndParameters
   H1-InclJet-HERAII  datafiles/h1/1406.4709/H1-HERAII-HighQ2-InclJets.dat  fastNLO          # ptmin>=11 q2min>333
   H1-Dijets-HERAII   datafiles/h1/1406.4709/H1-HERAII-HighQ2-Dijets.dat   fastNLO          # ptmin>=11 q2min>333
}}

############################################################
#  Specify Alpos theory (in 'AlposTheory' namespace)
############################################################
AlposTheory {{{
   # ------------------------------------------------------------#
   #   Specify used functions.. Give them a 'name' and specify the class
   # ------------------------------------------------------------#
   InitFunctions {{                                              # Specify all needed alpos functions (name,type)
     FunctionName           FunctionType                         # one line here! (header of 'table')
     CRunDec                CRunDec                              # Use name;type for convenience reason if only one instance of a function is used
     LHAPDF6                LHAPDF6                              # Use name;type
     MSTW                   LHAPDF6                              # Use name;type
   }}

   # ------------------------------------------------------------#
   # Useful parameter shorthand notations
   # ------------------------------------------------------------#
   iOrd                     1                                    # iOrd
   AlphasMz                 0.1180                               # Alpha_s
   mZ                       91.1876                              # mZ

   # ------------------------------------------------------------#
   #  Theory 'defaults'
   # ------------------------------------------------------------#
   CRunDec.AlphasMz         AlphasMz
   CRunDec.Mz               mZ
   CRunDec.nFlavor          5
   CRunDec.nLoop            2
   CRunDec.mur              100                                  # default! Only used for internal calculation.
   # ------------------------------------------------------------#
   LHAPDF6.LHAPDFFile       NNPDF31_nnlo_as_0118                 # Chose a PDF: CT10,  MSTW2008nlo_asmzrange, NNPDF30_as_nlo_0118
   LHAPDF6.PDFSet           0                                    # PDFSet of the given file
   LHAPDF6.xp               0.01                                 # default! Only used for internal calculation.
   LHAPDF6.muf              100                                  # default! Only used for internal calculation.

   # ------------------------------------------------------------#
   # fastNLO default values (Many values often overwritten by data steering)
   fastNLO.Filename         table.tab                            # default value! Always overwritten by data steering
   fastNLO.ScaleFacMuR      1                                    # default value! Often overwritten by data steering
   fastNLO.ScaleFacMuF      1                                    #
   fastNLO.Units            1                                    # 0: absoluteUnits, 1: PublicationUnits
   fastNLO.iOrd             iOrd                                 #
   fastNLO.MuRFuncForm      3                                    # 0: scale1, 1: scale2, 2: quad.sum, 3: quad.mean (partially overwritten by datasteering)
   fastNLO.MuFFuncForm      3                                    # 0: scale1, 1: scale2, 2: quad.sum, 3: quad.mean (partially overwritten by datasteering)
   fastNLO.PDF              LHAPDF6                              # PDF function
   fastNLO.Alpha_s          CRunDec                              # Alpha_s function

   # ------------------------------------------------------------#
   #  Theory 'specializations'
   # ------------------------------------------------------------#
   MSTW.LHAPDFFile          MSTW2008nlo_asmzrange
   #MSTW.PDFSet             0                                    # PDFSet of the given file

   H1-InclJet-HERAII.PDF    MSTW                                 # take th MSTW instance of LHAPDF as PDF

}}} # end of 'AlposTheory' namespace
# ------------------------------------------------------------#
```

Too small…

I'll remove 'comments'
→ next slide

```
DataTheorySets {{
   AlposName           SteerFile                                              TheoryFunction    CutsAndParameters
   H1-InclJet-HERAII   datafiles/h1/1406.4709/H1-HERAII-HighQ2-InclJets.dat    fastNLO           # ptmin>=11 q2min>333
   H1-Dijets-HERAII    datafiles/h1/1406.4709/H1-HERAII-HighQ2-Dijets.dat     fastNLO           # ptmin>=11 q2min>333
}}

#  Specify Alpos theory (in 'AlposTheory' namespace)
AlposTheory {{{
   # --- Specify used functions.. Give them a 'name' and specify the class
   InitFunctions {{                                          # Specify all needed alpos functions (name,type)
     FunctionName            FunctionType                    # one line here! (header of 'table')
     CRunDec                 CRunDec                         # Use name;type for convenience reason if only one
     LHAPDF6                 LHAPDF6                         # Use name;type
     MSTW                    LHAPDF6                         # Use name;type
   }}

   # --- Useful parameter shorthand notations
   iOrd                      1                               # iOrd
   AlphasMz                  0.1180                          # Alpha_s
   mZ                        91.1876                         # mZ

   #  --- Theory 'defaults'
   CRunDec.AlphasMz          AlphasMz
   CRunDec.Mz                mZ
   CRunDec.nFlavor           5
   CRunDec.nLoop             2
   CRunDec.mur               100                             # default! Only used for internal calculation.
   # ------------------------------------------------------------#
   LHAPDF6.LHAPDFFile        NNPDF31_nnlo_as_0118            # Chose a PDF: CT10,  MSTW2008nlo_asmzrange, NNPDF3(
   LHAPDF6.PDFSet            0                               # PDFSet of the given file
   LHAPDF6.xp                0.01                            # default! Only used for internal calculation.
   LHAPDF6.muf               100                             # default! Only used for internal calculation.

   # --- fastNLO default values (Many values often overwritten by data steering)
   fastNLO.Filename          table.tab                       # default value! Always overwritten by data steering
   fastNLO.ScaleFacMuR       1                               # default value! Often overwritten by data steering
   fastNLO.iOrd              iOrd                            #
   fastNLO.MuRFuncForm       3                               # 0: scale1, 1: scale2, 2: quad.sum, 3: quad.mean (
   fastNLO.PDF               LHAPDF6                         # PDF function
   fastNLO.Alpha_s           CRunDec                         # Alpha_s function

   # --- Theory 'specializations'
   MSTW.LHAPDFFile           MSTW2008nlo_asmzrange
   #MSTW.PDFSet               0                               # PDFSet of the given file

   H1-InclJet-HERAII.PDF     MSTW                            # take the MSTW instance of LHAPDF as PDF

}}} # end of 'AlposTheory' namespace
```

Two data sets: both use 'fastNLO' for their predictions

Functions

Function 'name'
Function 'type'

These parameter are finally 'the same'

Every function needs to specify its default input parameters

Parameters

A function is input to another function

Function 'type'

What Alpos has in memory

Functions

- **CRunDec**
  CRunDec
- **fastNLO**
  H1-Dijets-HERAII
- **AData**
  H1-Dijets-HERAII_Data
- **fastNLO**
  H1-InclJets-HERAII
- **AData**
  H1-InclJets-HERAII_Data
- **LHAPDF6**
  LHAPDF6
- **LHAPDF6**
  MSTW
- **SuperData**
  SuperData
- **SuperTheory**
  SuperTheory

Parameters (functions) and their return values

```
+  AlphasMz                                                                  0.118
+  CRunDec                                                                   0.116381
+  CRunDec.AlphasMz  --> AlphasMz                                            0.118
+  CRunDec.Mz  --> mZ                                                        91.1876
+  CRunDec.mur                                                               100
+  CRunDec.nFlavor                                                           5
+  CRunDec.nLoop                                                             2
+  H1-Dijets-HERAII                                                          71.7011 [0],  32.0377 [1],  7.15568 [2],  ... , 0.298477 [23];
+  H1-Dijets-HERAII.Alpha_s  --> CRunDec                                     0.116381
+  H1-Dijets-HERAII.Filename                                                 theoryfiles/fnh5001_60G_HS12.tab
+  H1-Dijets-HERAII.MuFFuncForm                                              0
+  H1-Dijets-HERAII.MuRFuncForm                                              3
+  H1-Dijets-HERAII.PDF  --> LHAPDF6                                         0 [0],  0.231235 [1],  0.339504 [2],  ... , 0 [12];
+  H1-Dijets-HERAII.ScaleFacMuF  --> fastNLO.ScaleFacMuF                     1
+  H1-Dijets-HERAII.ScaleFacMuR  --> fastNLO.ScaleFacMuR                     1
+  H1-Dijets-HERAII.Units                                                    0
+  H1-Dijets-HERAII.iOrd  --> iOrd  (const)                (const)           1
+  H1-Dijets-HERAII.iThr  --> fastNLO.iThr  (const)        (const)           -1
+  H1-Dijets-HERAII_Data  (const)                          (const)           70.5817 [0],  30.9722 [1],  8.0735 [2],  ... , 0.3087 [23];
+  H1-InclJet-HERAII                                                         70.9397 [0],  31.7108 [1],  7.13522 [2],  ... , 0.300239 [23];
+  H1-InclJet-HERAII.Alpha_s  --> CRunDec                                    0.116381
+  H1-InclJet-HERAII.Filename                                                theoryfiles/fnh5001_60G_HS12.tab
+  H1-InclJet-HERAII.MuFFuncForm                                             0
+  H1-InclJet-HERAII.MuRFuncForm                                             3
+  H1-InclJet-HERAII.PDF  --> MSTW                                           0 [0],  0.23321 [1],  0.35813 [2],  ... , 0 [12];
+  H1-InclJet-HERAII.ScaleFacMuF  --> fastNLO.ScaleFacMuF                    1
+  H1-InclJet-HERAII.ScaleFacMuR  --> fastNLO.ScaleFacMuR                    1
+  H1-InclJet-HERAII.Units                                                   0
+  H1-InclJet-HERAII.iOrd  --> iOrd  (const)               (const)           1
+  H1-InclJet-HERAII.iThr  --> fastNLO.iThr  (const)       (const)           -1
+  H1-InclJet-HERAII_Data  (const)                         (const)           70.5817 [0],  30.9722 [1],  8.0735 [2],  ... , 0.3087 [23];
+  LHAPDF6                                                                   0 [0],  0.231235 [1],  0.339504 [2],  ... , 0 [12];
+  LHAPDF6.LHAPDFFile                                                        NNPDF31_nnlo_as_0118
+  LHAPDF6.PDFSet                                                            0
+  LHAPDF6.muf                                                               100
+  LHAPDF6.xp                                                                0.01
+  MSTW                                                                      0 [0],  0.23321 [1],  0.35813 [2],  ... , 0 [12];
+  MSTW.LHAPDFFile                                                           MSTW2008nlo_asmzrange
+  MSTW.PDFSet  --> LHAPDF6.PDFSet                                           0
+  MSTW.muf  --> LHAPDF6.muf                                                 100
+  MSTW.xp  --> LHAPDF6.xp                                                   0.01
+  SuperData                                                                 70.5817 [0],  30.9722 [1],  8.0735 [2],  ... , 0.3087 [47];
+  SuperData.H1-Dijets-HERAII  --> H1-Dijets-HERAII_Data  (const)  (const)  70.5817 [0],  30.9722 [1],  8.0735 [2],  ... , 0.3087 [23];
+  SuperData.H1-InclJet-HERAII  --> H1-InclJet-HERAII_Data  (const)  (const)  70.5817 [0],  30.9722 [1],  8.0735 [2],  ... , 0.3087 [23];
+  SuperTheory                                                               71.7011 [0],  32.0377 [1],  7.15568 [2],  ... , 0.300239 [47];
+  SuperTheory.H1-Dijets-HERAII  --> H1-Dijets-HERAII                        71.7011 [0],  32.0377 [1],  7.15568 [2],  ... , 0.298477 [23];
+  SuperTheory.H1-InclJet-HERAII  --> H1-InclJet-HERAII                      70.9397 [0],  31.7108 [1],  7.13522 [2],  ... , 0.300239 [23];
+  fastNLO.Alpha_s  --> CRunDec                                              0.116381
+  fastNLO.Filename                                                          table.tab
+  fastNLO.MuFFuncForm                                                       3
+  fastNLO.MuRFuncForm                                                       3
+  fastNLO.PDF  --> LHAPDF6                                                  0 [0],  0.231235 [1],  0.339504 [2],  ... , 0 [12];
+  fastNLO.ScaleFacMuF                                                       1
+  fastNLO.ScaleFacMuR                                                       1
+  fastNLO.Units                                                             1
+  fastNLO.iOrd  --> iOrd  (const)                         (const)           1
+  fastNLO.iThr  (const)                                   (const)           -1
+  iOrd  (const)                                           (const)           1
+  mZ                                                                        91.1876
```

This is a single parameter instance, but it has two names

AlposTheory can become extensive: H1PDF2017 fit had 586 parameters

What Alpos has in memory

Parameters (functions) and their return values

Input parameters and input functions to a specific calculation

Return values (`fValue`) of parameter/function for the present 'TheorySet'

`SuperData`, `SuperTheory`: dedicated 'functions' which just concatenate fValues from their input parameters/functions

```
+  AlphasMz                                                    0.118
+  CRunDec                                                     0.116381
+  CRunDec.AlphasMz  --> AlphasMz                              0.118
+  CRunDec.Mz  --> mZ                                          91.1876
+  CRunDec.mur                                                 100
+  CRunDec.nFlavor                                             5
+  CRunDec.nLoop                                               2
+  H1-Dijets-HERAII                                            71.7011 [0],  32.0377 [1],  7.15568 [2],  ... , 0.298477 [23];
+  H1-Dijets-HERAII.Alpha_s  --> CRunDec                       0.116381
+  H1-Dijets-HERAII.Filename                                   theoryfiles/fnh5001_60G_HS12.tab
+  H1-Dijets-HERAII.MuFFuncForm                                0
+  H1-Dijets-HERAII.MuRFuncForm                                3
+  H1-Dijets-HERAII.PDF  --> LHAPDF6                           0 [0],  0.231235 [1],  0.339504 [2],  ... , 0 [12];
+  H1-Dijets-HERAII.ScaleFacMuF  --> fastNLO.ScaleFacMuF       1
+  H1-Dijets-HERAII.ScaleFacMuR  --> fastNLO.ScaleFacMuR       1
+  H1-Dijets-HERAII.Units                                      0
+  H1-Dijets-HERAII.iOrd  --> iOrd  (const)              (const) 1
+  H1-Dijets-HERAII.iThr  --> fastNLO.iThr  (const)      (const) -1
+  H1-Dijets-HERAII_Data  (const)                        (const) 70.5817 [0],  30.9722 [1],  8.0735 [2],  ... , 0.3087 [23];
+  H1-InclJet-HERAII                                           70.9397 [0],  31.7108 [1],  7.13522 [2],  ... , 0.300239 [23];
+  H1-InclJet-HERAII.Alpha_s  --> CRunDec                      0.116381
+  H1-InclJet-HERAII.Filename                                  theoryfiles/fnh5001_60G_HS12.tab
+  H1-InclJet-HERAII.MuFFuncForm                               0
+  H1-InclJet-HERAII.MuRFuncForm                               3
+  H1-InclJet-HERAII.PDF  --> MSTW                             0 [0],  0.23321 [1],  0.35813 [2],  ... , 0 [12];
+  H1-InclJet-HERAII.ScaleFacMuF  --> fastNLO.ScaleFacMuF      1
+  H1-InclJet-HERAII.ScaleFacMuR  --> fastNLO.ScaleFacMuR      1
+  H1-InclJet-HERAII.Units                                     0
+  H1-InclJet-HERAII.iOrd  --> iOrd  (const)             (const) 1
+  H1-InclJet-HERAII.iThr  --> fastNLO.iThr  (const)     (const) -1
+  H1-InclJet-HERAII_Data  (const)                       (const) 70.5817 [0],  30.9722 [1],  8.0735 [2],  ... , 0.3087 [23];
+  LHAPDF6                                                     0 [0],  0.231235 [1],  0.339504 [2],  ... , 0 [12];
+  LHAPDF6.LHAPDFFile                                          NNPDF31_nnlo_as_0118
+  LHAPDF6.PDFSet                                              0
+  LHAPDF6.muf                                                 100
+  LHAPDF6.xp                                                  0.01
+  MSTW                                                        0 [0],  0.23321 [1],  0.35813 [2],  ... , 0 [12];
+  MSTW.LHAPDFFile                                             MSTW2008nlo_asmzrange
+  MSTW.PDFSet  --> LHAPDF6.PDFSet                             0
+  MSTW.muf  --> LHAPDF6.muf                                   100
+  MSTW.xp  --> LHAPDF6.xp                                     0.01
+  SuperData                                                   70.5817 [0],  30.9722 [1],  8.0735 [2],  ... , 0.3087 [47];
+  SuperData.H1-Dijets-HERAII  --> H1-Dijets-HERAII_Data  (const)  (const)  70.5817 [0],  30.9722 [1],  8.0735 [2],  ... , 0.3087 [23];
+  SuperData.H1-InclJet-HERAII  --> H1-InclJet-HERAII_Data  (const)  (const)  70.5817 [0],  30.9722 [1],  8.0735 [2],  ... , 0.3087 [23];
+  SuperTheory                                                 71.7011 [0],  32.0377 [1],  7.15568 [2],  ... , 0.300239 [47];
+  SuperTheory.H1-Dijets-HERAII  --> H1-Dijets-HERAII          71.7011 [0],  32.0377 [1],  7.15568 [2],  ... , 0.298477 [23];
+  SuperTheory.H1-InclJet-HERAII  --> H1-InclJet-HERAII        70.9397 [0],  31.7108 [1],  7.13522 [2],  ... , 0.300239 [23];
+  fastNLO.Alpha_s  --> CRunDec                                0.116381
+  fastNLO.Filename                                            table.tab
+  fastNLO.MuFFuncForm                                         3
+  fastNLO.MuRFuncForm                                         3
+  fastNLO.PDF  --> LHAPDF6                                    0 [0],  0.231235 [1],  0.339504 [2],  ... , 0 [12];
+  fastNLO.ScaleFacMuF                                         1
+  fastNLO.ScaleFacMuR                                         1
+  fastNLO.Units                                               1
+  fastNLO.iOrd  --> iOrd  (const)                       (const) 1
+  fastNLO.iThr  (const)                                 (const) -1
+  iOrd  (const)                                         (const) 1
+  mZ                                                          91.1876
```

Technicalities V: data file

```
Data {{
  Q2_min   Q2_max  Pt_min  Pt_max    Sigma  stat.(%)  sys.(%)   had  had.err  JES(up)  JES(dn)    Model
     150      200       7      11  70.5817       2.7      2.9  0.93      2.2     0.94    -1.13     0.94
     150      200      11      18  30.9722       4.1      4.4  0.97      1.7     2.37    -2.54     0.56
     150      200      18      30   8.0735       6.4      5.3  0.96      1.1     3.36    -3.38     0.31
     150      200      30      50   0.9184      15.3     12.9  0.95      0.7     4.87    -5.28     0.16
     # [...]
}}

Subsets { Q2_min Pt_min }

Cuts { }

ErrorSet        "H1Hera-II"   # errors are labelled as "<ErrorSet>_<Name>", and are correlated with other of
ErrorUnit         Percent     # Are errors given as: "Absolute", "Relative", "Percent"

Errors {{
ErrorName      Column         Correlation
  Stat      stat.(%)             ESMC
  JES      JES(up):JES(dn)       EYM1
  Lumi            5              EYM1
  NP            had.err          TYM0.5
}}


TheoryFactors { had EW }

Stat_Matrix_Format        Matrix        # "Matrix" or "SingleValues" or specify single value only
Stat_Matrix_Type          Correlation   # 'Covariance', 'Correlation' or 'CorrelationPercent'

Stat_Matrix {{
  #
   1.00
  -0.20   1.00
  -0.11   0.02   1.00
  -0.02  -0.01   0.06   1.00
  -0.14   0.04   0.01   0.00   1.00
}}
```

A table called 'Data' with arbitrary colums
One colum named 'Sigma' !

Specifications of uncertainties:

1) Specify name of uncertaity source
2) Specify colum(s) of 'Data'
3) Specify 'details' on that uncertainty source

E/T: exp. or th. uncert.
S/T: stat. or syst. uncert.
M/A: mult. or additive uncert.
0–1/C: fraction of correlation, or dedicated matrix

Uncertainties and χ^2 calculation

```
double AChisqCov::DoEval(const double *p) const {
   //! Calculate chisq
   //!    chisq = sum_ij [m_i - t_i] * V^-1_ij * [m_i - t_i]
   //! V is the covariance matrix containing all uncertainties
   //!
   //! \note any uncertainties marked as multiplicative will
   //!       be treated as additive by this chi-square definition

   // --- set new theory parameters
   for ( unsigned int ipar = 0 ; ipar < fFitPar.size() ; ipar++ )
      SET_ANY(fFitPar[ipar],p[ipar],0);

   // --- get data and theory values
   const vector<double>& th = Theo()->GetValues();
   const vector<double>& da = Data()->GetValues();

   // --- get uncertainties from data and theory objects

   // additive errors can be considered directly
   const TMatrixDSym* InvCov = &AInvMatrices::Instance()->GetInvMatrix({
         &fData->GetSumErrorMatrix("AA", "AbsAvTotAll"),
         &fTheo->GetSumErrorMatrix("AA", "AbsAvTotAll")
      });

   // --- loop over all data points and calculate chisq
   double chisq = 0;
   for ( unsigned int x = 0 ; x<th.size() ; x++ ) {
      chisq += pow((da[x]-th[x]),2) * (*InvCov)[x][x] ;
      //cout<<"dt= "<<da[x]<<"\tth= "<<th[x]<<"\terr= "<<InvCov[x][x] <<endl;
      for ( unsigned int y = 0 ; y<x ; y++ ) {
         chisq += 2 * (da[x]-th[x]) * (*InvCov)[x][y] * (da[y]-th[y]);
      }
   }

   return chisq;
}
```

Example 1 (out of 23) χ^2 calculations

Set the parameter for the new iteration
(if it is called by some minimizer)

Retrieve updated predictions

Outstanding feature !!
1) any parameters are changed to the predictions, anywhere in the code
2) a line later, the respective prediction is obtaine in a maximal optimised way!

Access to uncertaintes: matrices

```
 actype = 'XY'
 X:    E/T/A:   experimental/theo. uncert./ all
 Y:    S/Y/A:   Stat./syst./all

 access = 'AaaBb[Ccc[Ddd]]' with:
 Aaa:  'Abs', 'Rel'       Obtain relative or absolute uncertainty
 Bb:   'Up', 'Dn', 'Av'   Obtain up-,down- or averaged uncertainty
 Ccc:  'Tot', 'Cor', 'Unc' Obtain total,corr., or uncorrelated part
 Ddd:  'Mul', 'Add', 'All' Obtain only additive or multiplicative
errors (or all [default])
```

… or vectors (not shown here)

Uncertainties and χ^2 calculation

Example 1 (out of 23) χ^2 calculations

```
//!
//! \note any uncertainties marked as multiplicative will
//!       be treated as additive by this chi-square definition

// --- set new theory parameters
for ( unsigned int ipar = 0 ; ipar < fFitPar.size() ; ipar++ )
   SET_ANY(fFitPar[ipar],p[ipar],0);

// --- get data and theory values
const vector<double>& th = Theo()->GetValues();
const vector<double>& da = Data()->GetValues();

// --- get uncertainties from data and theory objects
```

Set the parameter for the new iteration
(if it is called by some minimizer)

Retrieve updated predictions

Outstanding feature !!
1) any parameters are changed to the predictions, anywhere in the code
2) a line later, the respective prediction is obtaine in a maximal optimised way!

```
#define SET_ANY(X, VAL, ERR)    TheoryHandler::Handler()->GetParmD(X)->SetValue(VAL,ERR,false);
```

A very simple fit

```
# ------------------------------------------------------------#
DataTheorySets {{
  AlposName                   SteerFile                          TheoryFunctio
   mZ-PDG                     datafiles/pdg/mZ.dat               SingleConstan

}}

# ------------------------------------------------------------#
Tasks {{
   TaskType          TaskName                                  # first line i
   AFitter           MzFit                                     # Task
}}

# ------------------------------------------------------------#
MzFit {{{
   Minimizer                  TMinuit
   PrintLevel                 3
   Tolerance                  1
   Strategy                   1
   Chisq                      LogNormal #HERAFitterLogDefault
   PrintResults               true
   PrintCovariance            true

   FitParameters { mZ }
}}}

# ------------------------------------------------------------#

# ------------------------------------------------------------#
AlposTheory {{{
   InitFunctions {{
   }}                                                          # Specify all
   mZ                          92.000                          # mZ
   SingleConstant.theConst     100
   mZ-PDG.theConst             mZ
}}} # end of 'AlposTheory' namespace
# ------------------------------------------------------------#
```

```
**    7 **HESSE       3e+05
**********
  START COVARIANCE MATRIX CALCULATION.
EIGENVALUES OF SECOND-DERIVATIVE MATRIX:
        1.0000e+00
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=1.56158e-10 FROM HESSE     STATUS=OK              5 CALLS          24
                    EDM=3.12253e-10    STRATEGY= 1      ERROR MATRIX ACCUR
 EXT PARAMETER                                   INTERNAL      INTERNAL
 NO.   NAME      VALUE            ERROR          STEP SIZE       VALUE
  1  mZ           9.11876e+01   2.10000e-03   2.17408e-05   9.11876e+01
EXTERNAL ERROR MATRIX.    NDIM=  25    NPAR=  1    ERR DEF=1
 4.410e-06
EXTERNAL ERROR MATRIX.    NDIM=   1    NPAR=  1    ERR DEF=1
 4.410e-06
 # INFO.    [AFitter::Execute] Info. AFitter::Execute(). Initial chisq: 148
 # INFO.    [AFitter::Execute] Info. AFitter::Execute(). Final Chisq:   1.5

****************************************
Minimizer is TMinuit / Migrad
MinFCN                    =  1.56158e-10
NDf                       =           0
Edm                       =  3.12253e-10
NCalls                    =          24
mZ                        =      91.1876   +/-   0.0021

****************************************
Printing nuisance parameters:
 # INFO.    [AChisqBase::PrintNuisanceParameters] No Nuisance parameters ha

Correlation Matrix:

                          mZ
mZ                         1
 # INFO.    [AFitter::execute] Setting fitted value 'mZ' to: 91.1876 +/- 0.

 # INFO.    [Alpos::DoTasks] Task 'MzFit' done successfully. Runtime 00:00:

------------------------------------------------------------------------------
 # INFO.    [Alpos::DoTasks] All tasks done [1/1].
```

xFitter vs. Alpos for HERAPDF1.0

PDF fit of type HERAPDF1.0

- Use QCDNUM for structure funtions (ZMVFNS)
- Use QCDNUM for PDF evolution

```
$ ./src/alpos tutorial/6.herapdf10.str
```

```
InitFunctions {{
   FunctionName          FunctionType
   QcdnumInit            QcdnumInit
   PDFQ0_HERA_10pts      PDFQ0_HERA_10pts
   QcdnumPDF             QcdnumPDF              # needed for 'SavePDF'
}}
```

```
QcdnumDISCS.QcdnumInit                  QcdnumInit # PDF, alpha_s is provided through QCDNUM
QcdnumDISCS.Mw                          80.385
```

Alpos

```
MIGRAD MINIMIZATION HAS CONVERGED.
 FCN=599.113 FROM MIGRAD    STATUS=CONVERGED    1013 CALLS        1014 TOTAL
                     EDM=3.99026e-06    STRATEGY= 1  ERROR MATRIX UNCERTAINTY   0.5 per cent
  EXT PARAMETER                                   STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1  PDFQ0_HERA_10pts.gB   3.15870e-01   3.26536e-02   1.12029e-04   1.14505e+00
   2  PDFQ0_HERA_10pts.gC   9.41984e+00   7.00901e-01   2.88391e-03  -8.52656e-02
   3  PDFQ0_HERA_10pts.uvB   7.06652e-01   1.85413e-02   2.83958e-05   2.05594e+00
   4  PDFQ0_HERA_10pts.uvC   5.12802e+00   2.42370e-01  -5.04500e-04  -2.39463e-01
   5  PDFQ0_HERA_10pts.uvE   1.05656e+01   2.27037e+00  -1.11086e-03   2.61973e-02
   6  PDFQ0_HERA_10pts.dvC   4.58635e+00   3.73857e-01   1.26045e-03  -6.57842e-02
   7  PDFQ0_HERA_10pts.UbarC   1.33662e+00   1.88350e-01   5.33714e-04  -1.14586e-01
   8  PDFQ0_HERA_10pts.DbarA   1.13812e-01   3.84916e-03  -4.22832e-07  -5.24836e+00
   9  PDFQ0_HERA_10pts.DbarB  -2.13218e-01   4.63145e-03  -2.60368e-06   5.64812e+00
  10  PDFQ0_HERA_10pts.DbarC   2.82604e+00   7.80673e-01  -1.95571e-03  -5.93404e-02
```

HERAFitter

```
MIGRAD WILL VERIFY CONVERGENCE AND ERROR MATRIX.
COVARIANCE MATRIX CALCULATED SUCCESSFULLY

FCN=   599.0083     FROM MIGRAD    STATUS=CONVERGED   1091 CALLS     1094 TOTAL
                    EDM=  0.92E-05    STRATEGY= 1      ERROR MATRIX ACCURATE
 EXT PARAMETER                                         STEP         FIRST
 NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
  2    Bg         0.31589       0.32342E-01   0.33667E-04   -1.0115
  3    Cg          9.4195       0.68932       0.49584E-03   0.71943E-01
 12    Buv        0.70666       0.18317E-01   0.20803E-04   -1.8531
 13    Cuv         5.1276       0.24026       0.17813E-03   0.22745
 15    Euv         10.564        2.2212       0.16083E-02  -0.24652E-01
 23    Cdv         4.5884       0.37389       0.66150E-03   0.55725E-01
 33    CUbar       1.3370       0.19037       0.36544E-03   0.10098
 41    ADbar      0.11380       0.38530E-02   0.58487E-05    6.1119
 42    BDbar     -0.21323       0.46307E-02   0.58352E-05   -6.0739
 43    CDbar       2.8255       0.75066       0.70510E-03   0.50264E-01
```

Also 'HERAPDF2.0' benchmark available

Incomplete list of 'functions'

AUserFunction.h
ATwoPomeronParams.h
ATwoPomeronModel.h
AStrowp1.h
ASingleConstant.h
AQcdnumPDF.h
AQcdnumInit.h
AQcdnumDISCS.h
AQcdnumDISCSEWFit.h
AQcdnumDDISCS.h
AQcdnumAlphas.h
APDFQ0_QcdnumExample.h
APDFQ0_LHAPDF.h
APDFQ0_HERAStyle.h
APDFQ0_HERA.h
APDFQ0_diff.h
APDFQ0_BiLog.h
ALhapdf6.h
ALhapdf6Alphas.h
AH1DPDF2006.h
AFunctionTF1.h
AFixedValues.h

AExampleFunction.h
AEprc.h
ADPDF.h
ACRunDecFunction.h
ABelleTDCPV.h
AApplgrid.h
AApfelxxReggeonCS.h
AApfelxxPDF.h
AApfelxxDISCS.h
AApfelxxDDISCS.h
AApfelxxAlphas.h
AApfel.h
AApfelDISCS.h
AApfelDISCSEWFit.h
AApfelDDISCS.h
AAlphaEmRun.h
fastNLOAlpos.h
fastNLOAlposDPDF.h
AfastNLORatio.h
AfastNLOnormDIS.h
AfastNLOnormDISalt.h
AfastNLOInterpolPDFasNormDIS.h
AfastNLOInterpolPDFas.h
AfastNLO.h
AfastNLODiffDIS.h
AfastNLOalt.h

Incomplete list of 'functions'

AUserFunction.h
ATwoPomeronParams.h
ATwoPomeronModel.h
AStrowp1.h
ASingleConstant.h
AQcdnumPDF.h
AQcdnumInit.h
AQcdnumDISCS.h
AQcdnumDISCSEWFit.h
AQcdnumDDISCS.h
AQcdnumAlphas.h
APDFQ0_QcdnumExample.h
APDFQ0_LHAPDF.h
APDFQ0_HERAStyle.h
APDFQ0_HERA.h
APDFQ0_diff.h
APDFQ0_BiLog.h
ALhapdf6.h
ALhapdf6Alphas.h
AH1DPDF2006.h
AFunctionTF1.h
AFixedValues.h

AExampleFunction.h
AEprc.h
ADPDF.h
ACRunDecFunction.h
ABelleTDCPV.h
AApplgrid.h
AApfelxxReggeonCS.h
AApfelxxPDF.h
AApfelxxDISCS.h
AApfelxxDDISCS.h
AApfelxxAlphas.h
AApfel.h
AApfelDISCS.h
AApfelDISCSEWFit.h
AApfelDDISCS.h
AAlphaEmRun.h
fastNLOAlpos.h
fastNLOAlposDPDF.h
AfastNLORatio.h
AfastNLOnormDIS.h
AfastNLOnormDISalt.h
AfastNLOInterpolPDFasNormDIS.h
AfastNLOInterpolPDFas.h
AfastNLO.h
AfastNLODiffDIS.h
AfastNLOalt.h

List of tasks

Wrapper of ROOT::TMinimizer

Constrained least square minimizer (C++)

Another minimzer

```
if      ( ttype == AExampleTask::fTaskType )       return new AExampleTask(tname);
else if ( ttype == AFitter::fTaskType )            return new AFitter(tname);//AFitter(tname,fSteerfile,&fResul
else if ( ttype == AApcalc::fTaskType )            return new AApcalc(tname);//AFitter(tname,fSteerfile,&fResul
else if ( ttype == AApcFitter::fTaskType )         return new AApcFitter(tname);//AFitter(tname,fSteerfile,&fRe
else if ( ttype == AConstLQFitter::fTaskType )     return new AConstLQFitter(tname);//AFitter(tname,fSteerfile,
else if ( ttype == APrintTheorySet::TaskType() )   return new APrintTheorySet(tname);
else if ( ttype == ASaveTheorySet::TaskType() )    return new ASaveTheorySet(tname);
else if ( ttype == APrintSteering::TaskType() )    return new APrintSteering(tname);
else if ( ttype == AStatAnalysis::fTaskType )      return new AStatAnalysis(tname);
else if ( ttype == APrintErrorSummary::fTaskType ) return new APrintErrorSummary(tname);
else if ( ttype == APrintDataTheory::fTaskType )   return new APrintDataTheory(tname);
else if ( ttype == AChi2FitPDFas::fTaskType )      return new AChi2FitPDFas(tname);
else if ( ttype == AChi2InterpolPDFas::fTaskType ) return new AChi2InterpolPDFas(tname);
else if ( ttype == AChi2Scan::fTaskType )          return new AChi2Scan(tname);
else if ( ttype == AContour::fTaskType )           return new AContour(tname);
else if ( ttype == AReplaceDataWithTheoryValues::fTaskType )          return new AReplaceDataWithTheoryValues(
else if ( ttype == AApcFitter::fTaskType )         return new AApcFitter(tname);
else if ( ttype == ASavePDFTGraph::fTaskType )     return new ASavePDFTGraph(tname);
else if ( ttype == ASaveDPDFTGraph::fTaskType )    return new ASaveDPDFTGraph(tname);
else if ( ttype == ASaveDataTheory::fTaskType )    return new ASaveDataTheory(tname);
else if ( ttype == AScaleUncertainty::fTaskType )  return new AScaleUncertainty(tname);
else if ( ttype == APDFUncer::fTaskType )          return new APDFUncer(tname);
else if ( ttype == ALHAPDFErrors::fTaskType )      return new ALHAPDFErrors(tname);
```

Minimizer based on polynomial/spline fit to profile likelihood

2D Contour plot (using AFitter)

Add PDF uncertainties to the 'predictions'

Write out LHAPDF grid with uncertaintes

Resume

Pro's of Alpos

- Very simple to implement new prediction
- Straight forward to implement new minimizer
  new χ^2 definition
- At any place in the code:
  parameters of the predictions can be changed, and 'updated' predictions can be obtain in an efficient way
- Flexible data format (definition of uncertainties)

Con's of Alpos

- Steering file may become difficult to understand
- Output is sometimes difficult to obtain
  → requires (additional) 'tasks'
- Despite the API has a fairly strict syntax, for an optimized code, and making use of inherent caching algorithm in Alpos, some code optimizations are sometimes required
- Fortran (non-object-oriented) code may be difficult to interface, although, there is 'standardised' way

Summary

Alpos is an object-oriented data-to-theory-comparison framework

- The 'theory module' consists of parameter- and function-objects inheriting from the same base class
- The input to functions, and the relation between parameters are specified in the steering
- At any place in the code:
  parameters of the predictions can be changed,
  and 'updated' predictions can be obtain in an efficient way

Full list of 'functions'

```
   AFuncD* ptr = NULL;
   if      ( functype == AExampleFunction::fFunctionName )  ptr = new AExampleFunction(funcname);
   else if ( functype == ACRunDecFunction::fFunctionName )  ptr = new ACRunDecFunction(funcname);
#if _CMAKE_FOUND_APFEL
   else if ( functype == AApfelInit::fFunctionName )        ptr = new AApfelInit(funcname);
   else if ( functype == AApfelPDF::fFunctionName )         ptr = new AApfelPDF(funcname);
   else if ( functype == AApfelAs::fFunctionName )          ptr = new AApfelAs(funcname);
   else if ( functype == AApfelQEDEvol::fFunctionName )     ptr = new AApfelQEDEvol(funcname);
   else if ( functype == AApfelDISCS::fFunctionName )       ptr = new AApfelDISCS(funcname);
   else if ( functype == AApfelDISCSEWFit::fFunctionName )  ptr = new AApfelDISCSEWFit(funcname);
   else if ( functype == AApfelDDISCS::fFunctionName )      ptr = new AApfelDDISCS(funcname);
#endif //_CMAKE_FOUND_APFEL
#if _CMAKE_FOUND_APFELxx
   else if ( functype == AApfelxxDISCS::fFunctionName )       ptr = new AApfelxxDISCS(funcname);
   else if ( functype == AApfelxxDDISCS::fFunctionName )      ptr = new AApfelxxDDISCS(funcname);
   else if ( functype == AApfelxxAlphas::fFunctionName )      ptr = new AApfelxxAlphas(funcname);
   else if ( functype == AApfelxxPDF::fFunctionName )         ptr = new AApfelxxPDF(funcname);
   else if ( functype == AApfelxxReggeonCS::fFunctionName )   ptr = new AApfelxxReggeonCS(funcname);
#endif //_CMAKE_FOUND_APFELxx
   else if ( functype == ADPDF::fFunctionName )             ptr = new ADPDF(funcname);
   else if ( functype == ALhapdf6::fFunctionName )          ptr = new ALhapdf6(funcname);
   else if ( functype == ALhapdf6Alphas::fFunctionName )    ptr = new ALhapdf6Alphas(funcname);
   else if ( functype == AStrowp1::fFunctionName )          ptr = new AStrowp1(funcname);
   else if ( functype == AH1DPDF2006::fFunctionName )       ptr = new AH1DPDF2006(funcname);
   //else if ( functype == AAlphasDependentPDF::fFunctionName)ptr = new AAlphasDependentPDF(funcname);
   else if ( functype == ABelleTDCPV::fFunctionName )          ptr = new ABelleTDCPV(funcname);
   else if ( functype == AfastNLO::fFunctionName )          ptr = new AfastNLO(funcname);
   else if ( functype == AfastNLOalt::fFunctionName )       ptr = new AfastNLOalt(funcname);
   else if ( functype == AfastNLODiffDIS::fFunctionName )   ptr = new AfastNLODiffDIS(funcname);
#if _CMAKE_FOUND_QCDNUM
   else if ( functype == AfastNLOnormDIS::fFunctionName )   ptr = new AfastNLOnormDIS(funcname);
   else if ( functype == AfastNLOnormDISalt::fFunctionName )   ptr = new AfastNLOnormDISalt(funcname);
   //else if ( functype == AfastNLODiffDIS::fFunctionName )   ptr = new AfastNLODiffDIS(funcname);
#endif //_CMAKE_FOUND_QCDNUM
   else if ( functype == AfastNLOInterpolPDFas::fFunctionName )       ptr = new AfastNLOInterpolPDFas(funcname);
#if _CMAKE_FOUND_QCDNUM
   else if ( functype == AfastNLOInterpolPDFasNormDIS::fFunctionName )      ptr = new AfastNLOInterpolPDFasNormDIS(funcname);
#endif //_CMAKE_FOUND_QCDNUM
   else if ( functype == AfastNLORatio::fFunctionName )     ptr = new AfastNLORatio(funcname);
#if _CMAKE_FOUND_APPLGRID
   else if ( functype == AApplgrid::fFunctionName )         ptr = new AApplgrid(funcname);
#endif //_CMAKE_FOUND_APPLGRID
   else if ( functype == ASuperTheory::fFunctionName )      ptr = new ASuperTheory(funcname); // should not be used in steering
   else if ( functype == ASuperData::fFunctionName )        ptr = new ASuperData(funcname); // should not be used in steering
   else if ( functype == ASubsetFunction::fFunctionName )   ptr = new ASubsetFunction(funcname); // should not be used in steering
   else if ( functype == ASubsetData::fFunctionName )       ptr = new ASubsetData(funcname); // should not be used in steering
   else if ( functype == AData::fFunctionName )             ptr = new AData(funcname); //< A data set
   else if ( functype == ASingleConstant::fFunctionName )   ptr = new ASingleConstant(funcname);
   else if ( functype == AFunctionTF1::fFunctionName )      ptr = new AFunctionTF1(funcname);
#if _CMAKE_FOUND_QCDNUM
   else if ( functype == AQcdnumInit::fFunctionName )       ptr = new AQcdnumInit(funcname);
   else if ( functype == AQcdnumAlphas::fFunctionName )     ptr = new AQcdnumAlphas(funcname);
   else if ( functype == AQcdnumPDF::fFunctionName )        ptr = new AQcdnumPDF(funcname);
   else if ( functype == AQcdnumDISCS::fFunctionName )      ptr = new AQcdnumDISCS(funcname);
   else if ( functype == AQcdnumDDISCS::fFunctionName )     ptr = new AQcdnumDDISCS(funcname);

   else if ( functype == AQcdnumDISCSEWFit::fFunctionName ) ptr = new AQcdnumDISCSEWFit(funcname);
#endif //_CMAKE_FOUND_QCDNUM
   else if ( functype == AAlphaEmRun::fFunctionName )       ptr = new AAlphaEmRun(funcname);
   else if ( functype == AEprc::fFunctionName )             ptr = new AEprc(funcname);
   else if ( functype == APDFQ0_LHAPDF::fFunctionName )     ptr = new APDFQ0_LHAPDF(funcname);
   else if ( functype == APDFQ0_QcdnumExample::fFunctionName )     ptr = new APDFQ0_QcdnumExample(funcname);
   else if ( functype == APDFQ0_HERAStyle::fFunctionName )  ptr = new APDFQ0_HERAStyle(funcname);
   else if ( functype == APDFQ0_diff::fFunctionName )       ptr = new APDFQ0_diff(funcname);
   else if ( functype == APDFQ0_HERA::fFunctionName )       ptr = new APDFQ0_HERA(funcname);
   else if ( functype == APDFQ0_BiLog::fFunctionName )      ptr = new APDFQ0_BiLog(funcname);
   else if ( functype == ATwoPomeronModel::fFunctionName )      ptr = new ATwoPomeronModel(funcname);
   else if ( functype == ATwoPomeronParams::fFunctionName )     ptr = new ATwoPomeronParams(funcname);
   else if ( functype == AUserFunction::fFunctionName )         ptr = new AUserFunction(funcname);
   else if ( functype == AFixedValues::fFunctionName )          ptr = new AFixedValues(funcname);
```

List of tasks

```
if      ( ttype == AExampleTask::fTaskType )       return new AExampleTask(tname);
else if ( ttype == AFitter::fTaskType )            return new AFitter(tname);//AFitter(tname,fSteerfile,&fResul
else if ( ttype == AApcalc::fTaskType )            return new AApcalc(tname);//AFitter(tname,fSteerfile,&fResul
else if ( ttype == AApcFitter::fTaskType )         return new AApcFitter(tname);//AFitter(tname,fSteerfile,&fRe
else if ( ttype == AConstLQFitter::fTaskType )     return new AConstLQFitter(tname);//AFitter(tname,fSteerfile,
else if ( ttype == APrintTheorySet::TaskType() )   return new APrintTheorySet(tname);
else if ( ttype == ASaveTheorySet::TaskType() )    return new ASaveTheorySet(tname);
else if ( ttype == APrintSteering::TaskType() )    return new APrintSteering(tname);
else if ( ttype == AStatAnalysis::fTaskType )      return new AStatAnalysis(tname);
else if ( ttype == APrintErrorSummary::fTaskType ) return new APrintErrorSummary(tname);
else if ( ttype == APrintDataTheory::fTaskType )   return new APrintDataTheory(tname);
else if ( ttype == AChi2FitPDFas::fTaskType )      return new AChi2FitPDFas(tname);
else if ( ttype == AChi2InterpolPDFas::fTaskType ) return new AChi2InterpolPDFas(tname);
else if ( ttype == AChi2Scan::fTaskType )          return new AChi2Scan(tname);
else if ( ttype == AContour::fTaskType )           return new AContour(tname);
else if ( ttype == AReplaceDataWithTheoryValues::fTaskType )          return new AReplaceDataWithTheoryValues
else if ( ttype == AApcFitter::fTaskType )         return new AApcFitter(tname);
else if ( ttype == ASavePDFTGraph::fTaskType )     return new ASavePDFTGraph(tname);
else if ( ttype == ASaveDPDFTGraph::fTaskType )    return new ASaveDPDFTGraph(tname);
else if ( ttype == ASaveDataTheory::fTaskType )    return new ASaveDataTheory(tname);
else if ( ttype == AScaleUncertainty::fTaskType )  return new AScaleUncertainty(tname);
else if ( ttype == APDFUncer::fTaskType )          return new APDFUncer(tname);
else if ( ttype == ALHAPDFErrors::fTaskType )      return new ALHAPDFErrors(tname);
```

List of χ^2 definitions

```
if      ( chisq == AChisqLogNormal::GetChisqName() ) return new AChisqLogNormal(par,data,theo);
else if ( chisq == AChisqLogNormalNuisance::GetChisqName() ) return new AChisqLogNormalNuisance(par,data,theo);
else if ( chisq == AChisqLogNormalNuisanceFit::GetChisqName() ) return new AChisqLogNormalNuisanceFit(par,data,theo);
else if ( chisq == AChisqLogNormalStatUncorr::GetChisqName() ) return new AChisqLogNormalStatUncorr(par,data,theo);
else if ( chisq == AChisqNormalLogNormal::GetChisqName() ) return new AChisqNormalLogNormal(par,data,theo);
// --- chisq without covariance matrices
else if ( chisq == AChisqSimple::GetChisqName() ) return new AChisqSimple(par,data,theo);
else if ( chisq == AChisqSimpleNuisanceMultFit::GetChisqName() ) return new AChisqSimpleNuisanceMultFit(par,data,theo);
else if ( chisq == AChisqSimpleNuisanceAddFit::GetChisqName() ) return new AChisqSimpleNuisanceAddFit(par,data,theo);
// else if ( chisq == AChisqSimpleNuisanceMult::GetChisqName() ) return new AChisqSimpleNuisanceMult(par,data,theo);
// else if ( chisq == AChisqSimpleNuisanceAdd::GetChisqName() ) return new AChisqSimpleNuisanceAdd(par,data,theo);
// --- chisq with covariance matrices and nuisance parameters
else if ( chisq == AChisqCov::GetChisqName() ) return new AChisqCov(par,data,theo);
else if ( chisq == AChisqCMS::GetChisqName() ) return new AChisqCMS(par,data,theo);
else if ( chisq == AChisqCovMult::GetChisqName() ) return new AChisqCovMult(par,data,theo);
else if ( chisq == AChisqCovStatUncorr::GetChisqName() ) return new AChisqCovStatUncorr(par,data,theo);
else if ( chisq == AChisqNuisanceAdd::GetChisqName() ) return new AChisqNuisanceAdd(par,data,theo);
else if ( chisq == AChisqNuisanceMult::GetChisqName() )  return new AChisqNuisanceMult(par,data,theo);
// else if ( chisq == AChisqNuisanceMultFit::GetChisqName() ) return new AChisqNuisanceMultFit(par,data,theo);
// else if ( chisq == AChisqNuisanceAddFit::GetChisqName() ) return new AChisqNuisanceAddFit(par,data,theo);
// --- Attempt for HERAFitter style chisq's
else if ( chisq == AChisqHERAFitterDefault::GetChisqName() ) return new AChisqHERAFitterDefault(par,data,theo);
else if ( chisq == AChisqHERAFitterLogDefault::GetChisqName() ) return new AChisqHERAFitterLogDefault(par,data,theo);
else if ( chisq == AChisqHERAFitterDefaultMatrix::GetChisqName() ) return new AChisqHERAFitterDefaultMatrix(par,data,theo);
else if ( chisq == AChisqHERAFitterDefaultFit::GetChisqName() ) return new AChisqHERAFitterDefaultFit(par,data,theo);
else if ( chisq == AChisqHERAFitterFull::GetChisqName() ) return new AChisqHERAFitterFull(par,data,theo);
else if ( chisq == AChisqHERAFitterFullImproved::GetChisqName() ) return new AChisqHERAFitterFullImproved(par,data,theo);
else if ( chisq == AChisqD0Fit::GetChisqName() ) return new AChisqD0Fit(par,data,theo);
else if ( chisq == AChisqD0StatCorrFit::GetChisqName() ) return new AChisqD0StatCorrFit(par,data,theo);
else if ( chisq == ALogLikelihood::GetChisqName() ) return new ALogLikelihood(par,data,theo);
else if ( chisq == APull::GetChisqName() ) return new APull(data,theo);
```