

---

---

# **xFitter: introduction and software status**

— S. Glazov, xFitter workshop,  
CERN, 2 May 2023 —

---

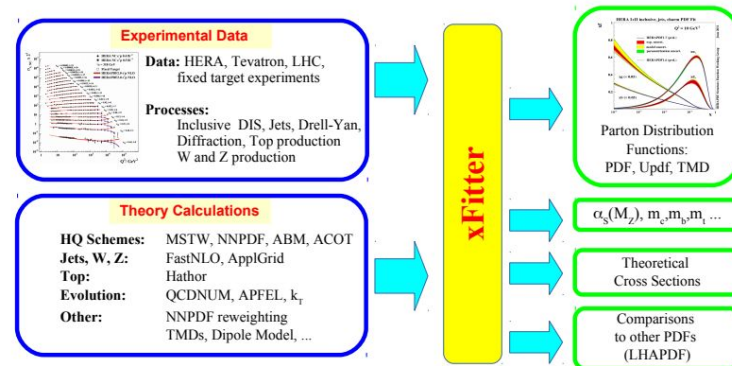
---

# Last workshop: Orsay March, 2022



- More than a year since the last face to face meeting that triggered a few developments

# xFitter overview



- xFitter is a QCD analysis tool
- Combines experimental data and theory with the focus on parton distribution function determination and other QCD parameters
- xFitter is supported by developers team, with a loose governance, consisting of experimentalists and theorists, with emphasis on phenomenology.
- xFitter developers continue software development of the package, support existing code, and also perform various analyses that are published as a team or as individual authors.

## chatGPT ([you.com](https://you.com)) knows it too:

Sure, xFitter is a free, open-source QCD fit framework that is used to extract parton distribution functions (PDFs) from experimental data. It is designed to perform global fits to various experimental data sets, which involve fitting theoretical models to the data to determine the PDFs that best describe the experimental observations. These fits are crucial in making predictions about high-energy particle physics processes and in testing the consistency of the Standard Model of particle physics.

xFitter can also be used to study the impact of various theoretical uncertainties on the PDFs and to produce nuclear PDFs. In addition, xFitter provides tools for PDF uncertainty propagation, PDF reweighting, and PDF correlation benchmarking, making it a versatile tool for studying PDFs in particle physics research.

Overall, xFitter is a powerful tool that can be used in a variety of research applications related to extracting PDFs and studying high-energy particle physics processes.

# xFitter organization

According to the search results, the xFitter project is developed by a team of researchers from various institutions and universities around the world. The xFitter developer's team includes Vittorio Bertone, Dieter Britzger, Stefano Camarda, Alexander Cooper-Sarkar, Amanda Cooper-Sarkar, Joey Huston, Pavel Nadolsky, Fred Olness, Mark Sutton, Oleksandr Zenaiev, and others. The xFitter project is an open-source package that aims to provide a framework for the determination of parton distribution functions (PDFs) and other QCD observables using global fits to various data sets <sup>1</sup>.

addi.ehu <sup>1</sup>

- xFitter developments are coordinated at biweekly xFitter developers' meetings (Wed 3pm CET/CEST).
- Communications occur using [xfitter-devel@desy.de](mailto:xfitter-devel@desy.de) mailing list (ask me if you want to subscribe)
- No formal "spokesperson" position
- Everybody welcome to join, however for signing developers' papers a proof of contribution to the project is required.

We had more strict governance model in the past, the current loose one seem to work  
Ok, but it is open for a change



# xFitter software overview



- Two supported “fortran-dominated” releases with a nuclear fork, downloadable from <https://www.xfitter.org/xFitter/xFitter/DownloadPage>
  - 2.0.0 Frozen frog
  - 2.0.1 Old Fashioned (mostly fixes for 2.0.0)
  - 2.0.1N Nuclear Daiquiri
- Several analyses used 2.1.0-release candidate, the code is tagged and can be found at gitlab <https://gitlab.cern.ch/fitters/xfitter/-/tags/master> before PionCeres merge
- Release 2.2.0 “Future Freeze” which contains several major updates.
- Several developments in master branch on gitlab, in particular dyturbo interface.

# xFitter code organization: gitlab and jira

The screenshot shows the GitLab interface for the 'xFitter' project. At the top, there's a navigation bar with the GitLab logo and a search bar. Below that, the project name 'xFitter' is displayed along with its ID '7780' and a 'Leave project' link. Statistics for the project are shown: 3,941 Commits, 77 Branches, 5 Tags, 1 MB Files, and 1.5 GB Storage. A merge request is visible, titled 'Merge branch 'clean-up-unused-variables' into 'master'', authored by Alexander Glazov 5 days ago. Below the merge request, there are buttons for 'README', 'Other', 'CI/CD configuration', 'Add Kubernetes cluster', and 'Configure Integrations'. At the bottom, a table lists files and their last commit information.

The screenshot shows the Jira issue tracking interface for the xFitter project. At the top, the version 'Version 2.2.0 UNRELEASED' is displayed, along with a 'Release' button. Below that, a progress bar indicates '30+ days overdue'. The interface displays the number of issues in various states: 43 Issues in version, 39 Issues done, 3 Issues in progress, and 1 Issue to do. A table lists the issues, including XFITTER-153, XFITTER-146, and XFITTER-148, all assigned to Alexander Glazov and in 'IN PROGRESS' status.

xFitter uses CERN services for the code <https://gitlab.cern.ch/fitters/xfitter> and issue tracking <https://its.cern.ch/jira/projects/XFITTER> . A CERN account is required for pull requests/ticket issue. A mirror <https://gitlab.com/fitters/xfitter> can be used without CERN account.

# xFitter documentation: gitlab wiki

filters > xfitter > Wiki > Home

Last edited by  **Simone Amoroso** 1 month ago

Page history

New page

## Home



## Version 2.2

### User

IF YOU HAVE A PROBLEM

[Installation script](#)

[Installing all datafiles](#)

[Installing with cmake](#)

### Program steering

[Basic program usage](#)

[YAML steering](#)

[Data file format](#)

[Chi2 definition](#)

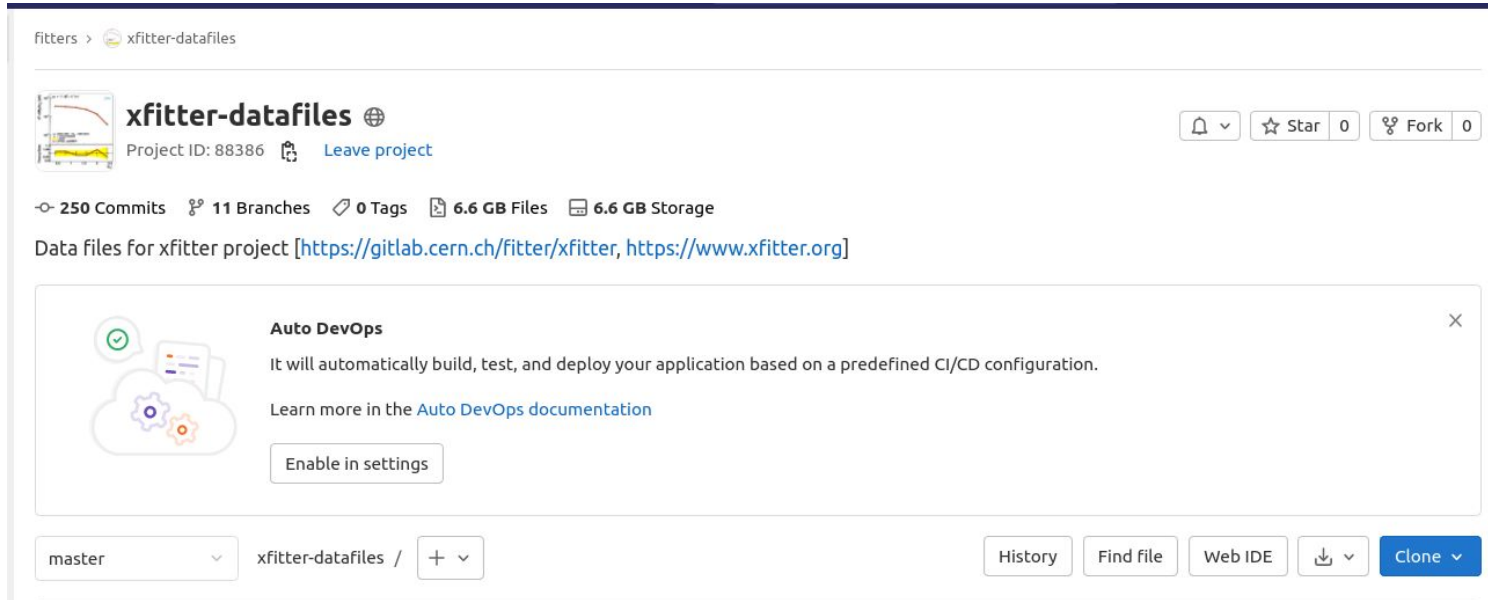
[Cuts](#)

Software documentation is moved to gitlab wiki <https://gitlab.cern.ch/filters/xfitter/-/wikis/home> which is significantly improved, should correspond to 2.2.0 / master branch

Please check the wiki before asking questions, and try to improve it.



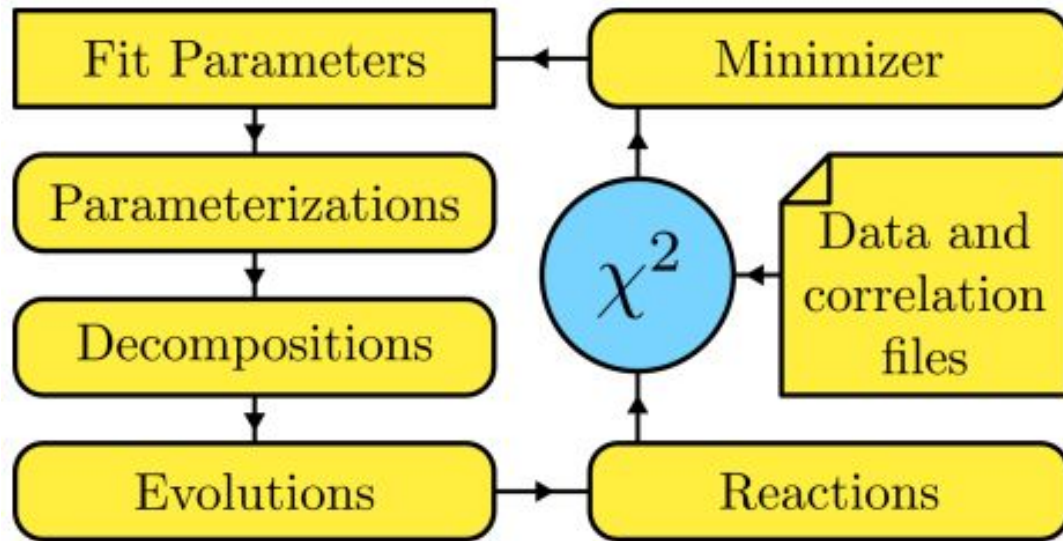
# xFitter data access: gitlab



The screenshot shows the GitLab interface for the repository 'xfitter-datafiles'. At the top, there's a breadcrumb 'fitters > xfitter-datafiles'. Below that, the repository name 'xfitter-datafiles' is displayed with a globe icon, followed by 'Project ID: 88386' and a 'Leave project' link. To the right, there are buttons for notifications, stars (0), and forks (0). Below this, statistics are shown: 250 Commits, 11 Branches, 0 Tags, 6.6 GB Files, and 6.6 GB Storage. A link provides data files for the xFitter project. A 'Auto DevOps' notification box is present, explaining that it will automatically build, test, and deploy applications based on predefined CI/CD configurations, with a link to the documentation and an 'Enable in settings' button. At the bottom, there are dropdown menus for 'master' and 'xfitter-datafiles /', and buttons for 'History', 'Find file', 'Web IDE', a download icon, and 'Clone'.

- Data access is now arranged via <https://gitlab.cern.ch/fitters/xfitter-datafiles>
- At the moment keep two versions of data files: compatible with 2.2.0 (“-thexp.dat”) and with 2.0.0.

# xFitter release 2.2.0



- Significant changes in the internal structure
- Re-written interfaces to minimizers, PDF parameterisation, decomposition, evolution and theory reactions
- Large changes in the user interface
- Data handling, format and chi2 calculation remain largely the same (but there are changes)

From talk by [Ivan Novikov](#)

# New reaction interface

```
class ReactionTheory{
public:
    ReactionTheory() {};
    virtual ~ReactionTheory() {};
public:
    using super=ReactionTheory;
    virtual string getReactionName()const=0; ///< Returns expected reaction name. Normally generated automatically by AddReaction.py
    virtual void atStart(); //called once after everything else is initialized
    virtual void atIteration(); //called in the beginning of each chi2 evaluation
    virtual void initTerm (TermData*);//called once for each term, after atStart()
    virtual void reinitTerm(TermData*);//called when some parameters for this term have changed and need to be re-read
    virtual void freeTerm (TermData*);//called for each term just before the ReactionTheory is destroyed. For cleanup
//The following 2 methods are TEMPORARY, poorly defined and probably will be replaced
    virtual void atFCN3();
    virtual void atMakeErrorBands(int i);
//! Main function to compute predictions for given term. Return results by filling val and errors
    virtual void compute(TermData*,valarray<double>&val,map<string,valarray<double> >&errors)=0;
};
```

- All theory predictions inherit from ReactionTheory class.
- Design is to have single instance of the class which handles multiple datafiles

# Updates of the program control

## parameters.yaml

Since version 2.2, parameters.yaml is the main steering file for xfitter. Some options that control the old fortran code are still in steering.txt, but we are planning to slowly migrate to the YAML steering and get rid of steering.txt completely.

The fitted parameters, the used parameterizations, decompositions and evolutions are defined in parameters.yaml. See [defining parameters](#), [defining parameterisations](#), [defining evolutions](#)

### Including files

Other YAML files can be included in the main file like this:

```
? !include PATH_TO_FILE
```

- Major changes in the way xFitter is steered
- Most of the parameters are moved to **parameters.yaml** file with only few options controlled by **steering.txt** namelists.
- The only namelists read from **steering.txt** are &InFiles, &InCorr, &CovarToNuisance, &xFitter (and only chi2 part of it), &Output, and &Cuts, the plan is to drop it altogether with version 2.3.0

# Future single configuration file (?)

can you give an example of yaml code in xfitter?

Certainly, here's an example of YAML code for a simple xFitter configuration file:

```
# Example xFitter configuration file
---
General:
# Fitting options
Fitting:
  Method: MINUIT
  Tolerance: 0.001
  MaxIterations: 1000

# Data options
Data:
  InputFiles:
    - datafile1.dat
    - datafile2.dat
  Format: ASCII
  Columns:
    x: 1
    y: 2
    error: 3

# PDF options
PDF:
  Driver: LHAPDF
  SetName: CT14nnlo

# Output options
Output:
  Directory: output/
  Filename: results.txt
  Format: ASCII
```

You.com thinks that we already got rid of steering.txt.

# Data format changes

```
!* File produced by HERAVerager
!* Created on 20160405

2.0.0 format

&Data
  Name = 'ATLAS W+ lepton rapidity 2011'
  Reaction = 'CC_pp'

  TheoryType = 'expression'
  TermName = 'A1', 'K'
  TermType = 'applgrid','kfactor'
  TermSource = 'datafiles/lhc/atlas/wzProduction/1612.03016/grid-40-6-15-3-Wplus_wyl.root',
              'datafiles/lhc/atlas/wzProduction/1612.03016/kf.wplus.txt'
  TheorExpr = 'K*A1/1000'

  NData = 11
  NColumn =138

-UU:----F1 wplus.dat Top L15 Git:master (Fortran company) -----

!* File produced by HERAVerager
!* Created on 20160405

2.2.0 format

&Data
  Name = 'ATLAS W+ lepton rapidity 2011'
  Reaction = 'CC_pp'

  TermName = 'A','K'
  TermSource = 'APPLgrid','KFactor'
  TermInfo = 'GridName=datafiles/lhc/atlas/wzProduction/1612.03016/grid-40-6-15-3-Wplus_wyl.root',
            'FileName=datafiles/lhc/atlas/wzProduction/1612.03016/kf.wplus.txt',FileColumn=3'
  TheorExpr = 'K*A/1000'

  NData = 11
  NColumn = 138

-UU:----F1 wplus-thexp.dat Top L5 Git:master (Fortran company) -----
```

- Data format remains largely the same, main changes in the description of the theory.
- TheoryType, Reaction, TermType parameters become optional (“Reaction” is still used for cuts)
- There is a change of the kFactor table format: becomes more flexible, but also not compatible.
- Dataset parameters can be given in TermInfo, they can be also modified in the



# Flexible evolution bindings

```
Evolutions:
  NAME:
    class: EVOLUTION_CLASS
    EVOLUTION_OPTIONS
  proton-QCDNUM:
    class: QCDNUM
    decomposition: proton
    #QCDNUM-specific options
    xGrid : [9.9e-7, 0.01, 0.1, 0.4, 0.7]
    xGridW : [1, 2, 4, 8, 16]
    Q2Grid : [1., 2.05e8 ]
    Q2GridW : [1., 1.]
    NQ2bins : 120
    NXbins : 200
    Read_QCDNUM_Tables : 1
    SplineOrder : 2
    ICheck : 0
  proton-LHAPDF:
    class: LHAPDF
    set: "CT10nlo"
    member: 0
  antiproton:
    class: FlipCharge
    input: proton-LHAPDF
  proton-APFELxx:
    ? !include evolutions/APFELxx.yaml
    decomposition: proton
  ...
```

- Similar to **ReactionTheory**, evolution codes are based on a **BaseEvolution** class
- Evolutions provide computations of the PDFs,  $\alpha_S$ , and other parameters based on **PDFdecomposition**, external input, or other evolutions.
- xFitter job can have several or no evolutions (for reactions such as TensorPomeron)
- The extended flexibility simplifies fits involving combination of several targets, e.g. pp, ppbar, pPb, etc.

# Changes in the minimizer

```
Parameters:
NAME: DEFINITION
Bg : [ -0.061953, 0.27 ]
Cg : [ 5.562367, 0.32 ]
Adv : DEPENDENT
Bdv : [ 1.029995, 0.06 ]
Cdv : [ 4.846279, 0.3 ]
Cubar: [ 7.059694, 0.8 ]
Dubar: [ 1.548098, 1.0 ]
Adbar: [ 0.1613, 0.01 ]
Bdbar: [ -0.1273, 0.004 ]
Cdbar: # another example of providing value, step etc.
  value: 9.586246
  step: 1.2345
  #min
  #max
  #pr_mean
  #pr_sigma
ZERO : 0
fs : 0.4 #no step means fixed
DbarToS: "=fs/(1-fs)"

Minimizer: MINUIT
MINUIT:
  Commands: |
    call fcn 1
    migrad
    hesse
    call fcn 3

doErrors : Hesse # or Pumpkin
```

- Parameters are now specified in **parameters.yaml** file, with the syntax loosely following the one from minuit
- Parameters can be also provided as functions of other parameters
- Parameters are then controlled by minimizers, two of which are interfaced: fortran Minuit and CERES.
- Significant improvement in CERES in master branch.
- More strict checks of the convergence compared to previous versions: **an update to that is in a branch.**

# From autotools to cmake

The build system has been completely rewritten using cmake. The new system is faster and more reliable.

Two libraries are required: **QCDNUM** and **yaml-cpp**. All other libraries are optional, cmake automatically detects whether they are installed and disables optional modules/features as necessary. After installing dependencies, one can use the wrapper script:

```
./make.sh install      - configure, compile, and install
./make.sh build        - configure and compile
./make.sh              - same (configure and compile)
./make.sh run          - configure, compile, install, and run
./make.sh clean        - delete all build files
./make.sh uninstall    - delete all installed files
./make.sh reconfigure  - configure from scratch
```

By default it builds in `./build` and installs in-source.

More information can be found on the wiki:

<https://gitlab.cern.ch/fitters/xfitter/-/wikis/Installation>

(from Ivan's slides [https://indico.desy.de/event/25055/contributions/55814/attachments/36157/45265/status-master\\_1.pdf](https://indico.desy.de/event/25055/contributions/55814/attachments/36157/45265/status-master_1.pdf))

# Improved validation

```
7220 Testing chi2scanMTOp ... PASS [details in temp/chi2scanMTOp/test.log]
7221 Testing defaultNLO ... PASS [details in temp/defaultNLO/test.log]
7222 Testing defaultNNLO ... PASS [details in temp/defaultNNLO/test.log]
7223 Testing evolutionAPFELxx ... PASS [details in temp/evolutionAPFELxx/test.log]
7224 Testing fractalFit ... PASS [details in temp/fractalFit/test.log]
7225 Testing modifyDataSetParameter ... PASS [details in temp/modifyDataSetParameter/test.log]
7226 Testing paramABMP16 ... PASS [details in temp/paramABMP16/test.log]
7227 Testing paramBG ... PASS [details in temp/paramBG/test.log]
7228 Testing profilerAs ... PASS [details in temp/profilerAs/test.log]
7229 Testing profilerLHAPDF ... PASS [details in temp/profilerLHAPDF/test.log]
7230 Testing profilerLHAPDF-HERAPDF20 ... PASS [details in temp/profilerLHAPDF-HERAPDF20/test.log]
7231 Testing ttbar3D ... PASS [details in temp/ttbar3D/test.log]
7232 -> 39 test(s) PASS
7233 ✓ Cleaning up project directory and file based variables 00:01
7235 Job succeeded
```

- Automatic validation of all pull requests / nightly builds
- Tests with minimal and full installation, including all dependences
- Most of functionality covered, plans to extend further.

# Important fixes since 2.2.0: gluon parameterisation

1. The functional form of the negative gluon term has changed

The canonical form is

$$xg(x) = A_g x^{B_g} (1-x)^{C_g} - A'_g x^{B'_g} (1-x)^{C'_g},$$

but the actual implementation was

$$xg(x) = A_g (x^{B_g} (1-x)^{C_g} - A'_g x^{B'_g} (1-x)^{C'_g}).$$

While these forms can express the same set of functions, the correlations between parameters are different and thus the shape of the  $\chi^2$  function changes. It turns out that the second form makes the fit more unstable.

The `NegativeGluon` parameterisation class has been changed to adapt the first form again. The sum rule is implemented accordingly. In all affected examples, the `Ag'` parameter has been adjusted to keep the PDFs unchanged. If necessary, the second form can still be obtained, using the `Expression` parameterisation.

Thanks to Florian for this fix [https://gitlab.cern.ch/fitters/xfitter/-/merge\\_requests/290](https://gitlab.cern.ch/fitters/xfitter/-/merge_requests/290)

# xFitter chi2 minimization problems

## Example 1

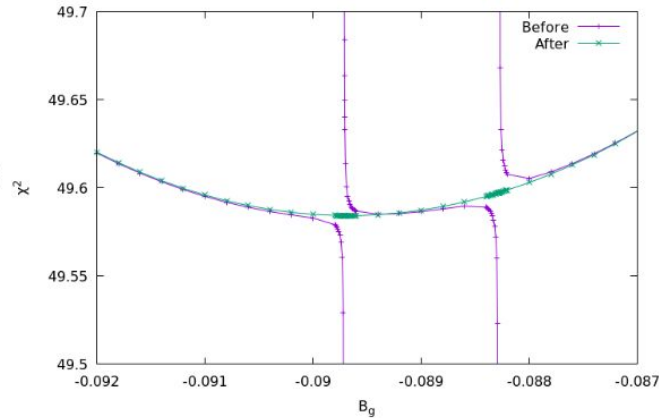
- ▶ Cross section diverges, leading to divergence of  $\chi^2$
- ▶ Problem is in calculation of  $F_L$  term in inclusive DIS cross sections in RT-scheme

$$F_{L,RT} = F_L \cdot \frac{F_{L,\gamma,RT}}{F_{L,\gamma}}$$

- ▶ If  $F_{L,\gamma} \rightarrow 0$ , this term diverges
- ▶ Temporary solution: apply correction additive (available in master since August)

$$F_{L,RT} = F_L + F_{L,\gamma,RT} - F_{L,\gamma}$$

- ▶ Better solution: improve theory implementation (work is in progress?)



- ▶ Plots:  $\chi^2$  scan of one parameter, while keeping all others constant
- ▶ Scan range is smaller than fit uncertainty → expect smooth line (not necessarily minimum)

Additive option included by Stefano in RT implementation since summer 2022.

[https://gitlab.cern.ch/fitters/xfitter/-/merge\\_requests/275](https://gitlab.cern.ch/fitters/xfitter/-/merge_requests/275)



# Parallel RT code

```
#RT_DISNC:                               # Reaction-specific settings
varin0 : 0.0
varin1 : 1.0
varin2 : -0.66666666666667
varin3 : 1.0                               # recommended RT OPT settings
additive: 1                               # "Additive" treatment of gamma-Z in
threads: 8                                # multi-thread support, how many th
```

```
glazov@starfive:~/xfitter$ time ./tools/test.sh CERES-fit
Testing CERES-fit ... PASS [details in temp/CERES-fit/test.log]
-> 1 test(s) PASS

real    24m30.764s
user    24m29.369s
sys     0m2.249s
glazov@starfive:~/xfitter$ time ./tools/test.sh CERES-parallel
Testing CERES-parallel ... PASS [details in temp/CERES-parallel/test.log]
-> 1 test(s) PASS

real    12m50.854s
user    34m12.718s
sys     1m12.124s
```

← Running on risc-V 4-core CPU

```
size_t Npr = Np/threads+1;
//std::cout << " Np " << Np << " Npr " << Npr << std::endl;
for (int P = 0; P < threads; P++)
{
    pid_t id = fork();
    if (id == 0)
    {
        close(fd[0]);
        for (size_t i = P*Npr; i < std::min(Np,(P+1)*Npr); i++)
        {
            if (!(q2[i] > 1.0))
                continue;
            stf fs;

            //std::cout << P << " " << i << std::endl;
            mstwnc_wrap_(x[i], q2[i], 1,
                        f2, f2c, f2b, fl, flc, flb,
                        iflag, i + 1, 1., 0.1, 0);
        }
    }
    pid_t wpid;
    while ((wpid = wait(&status)) > 0)
```

# Release strategy

Version 2.3.0 **UNRELEASED**

Start: 11/Nov/21 Release: 23/Dec/22 [Release Notes](#)

Polished 2.2.0, some extra features

Release

291 days left

9 Issues in version   0 Issues done   0 Issues in progress   9 Issues to do

1-9 of 9

[View in Issue Navigator](#)

P	T	Key	Summary	Assignee	Status
⊖	+	XFITTER-68	Theory reaction for Difftop fastNLO	Andrey Sapronov	OPEN
⋮	+	XFITTER-54	S-ACOT chi at NNLO	Fred Olness	OPEN
⋮	↑	XFITTER-139	Pure abstract base DIS classes	Alexander Glazov	OPEN
⋮	↑	XFITTER-143	Extend base parameterisation class to allow for analytic (automatic?) d...	Alexander Glazov	OPEN
=	↑	XFITTER-141	Improve CERES interface	Simone Amoroso	OPEN
=	↑	XFITTER-142	Extra simple parameterisations	Simone Amoroso	OPEN
⋮	⊖	XFITTER-127	for ceres minimizations, parameters are not reported at the end of the ...	Simone Amoroso	OPEN
⋮	+	XFITTER-25	New DYTURBO interface	Stefano Camarda	OPEN
⋮	↑	XFITTER-27	Update TMD code	Hannes Jung	OPEN

1-9 of 9

- 2.2.1 (?) could target issues discovered with 2.2.0
- Few items moved from 2.2.0 to 2.3.0
- Further changes are to be discussed, including major update of data interface and chi2 computation.

# Workshop goals

- xFitter software developments, decide on future release 2.3.0
- Feedback, bug fixes
- xFitter developers' team ongoing and potential future analyses.