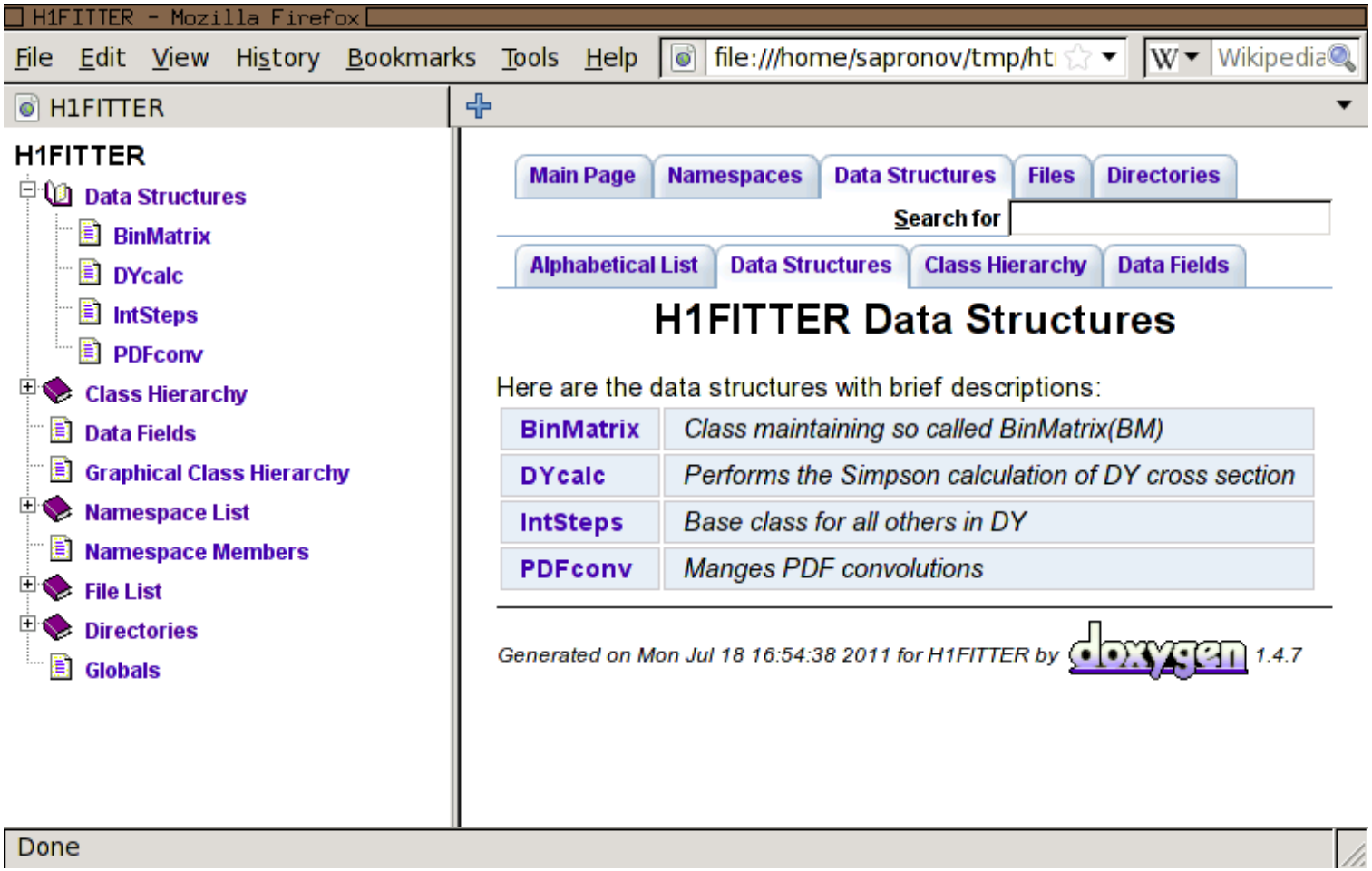
DY LO integration in HERAfitter.

Sapronov Andrey

February 14, 2012

DY LO integration code

The code is used for optimized born level Drell-Yan cross-section integration in LHC experiments. It supports custumized binnings and kinematic cuts.



DY integration code:

Simple LO cross section formulae: DY NC: $pp \to Z/\gamma \to e^+e^-$

$$\frac{d\sigma^2_\gamma}{dMdydcos\theta^*} = N_c C^2_{q\bar{q}} \frac{8\alpha^2}{3M^3}\tau \times \sum_q e_q^2 f_q(x_1, M) f_{\bar{q}}(x_2, M) F_{q\bar{q}}(1+\cos^2\theta^*, \cos\theta^*)$$

DY CC: $pp \to W^\pm \to e^\pm\nu$

$$\frac{d\sigma^3_{W^\pm}}{dMdydcos\theta^*} = \frac{\pi\alpha^2}{48 s_W^4} M\tau \frac{(1-cos\theta^*)^2}{(M^2-M_W^2)^2+\Gamma_W^2 M_W^2} \times \sum_{qq'} V_{qq'} f_q(x_1, M) f_{q'}(x_2, M)$$

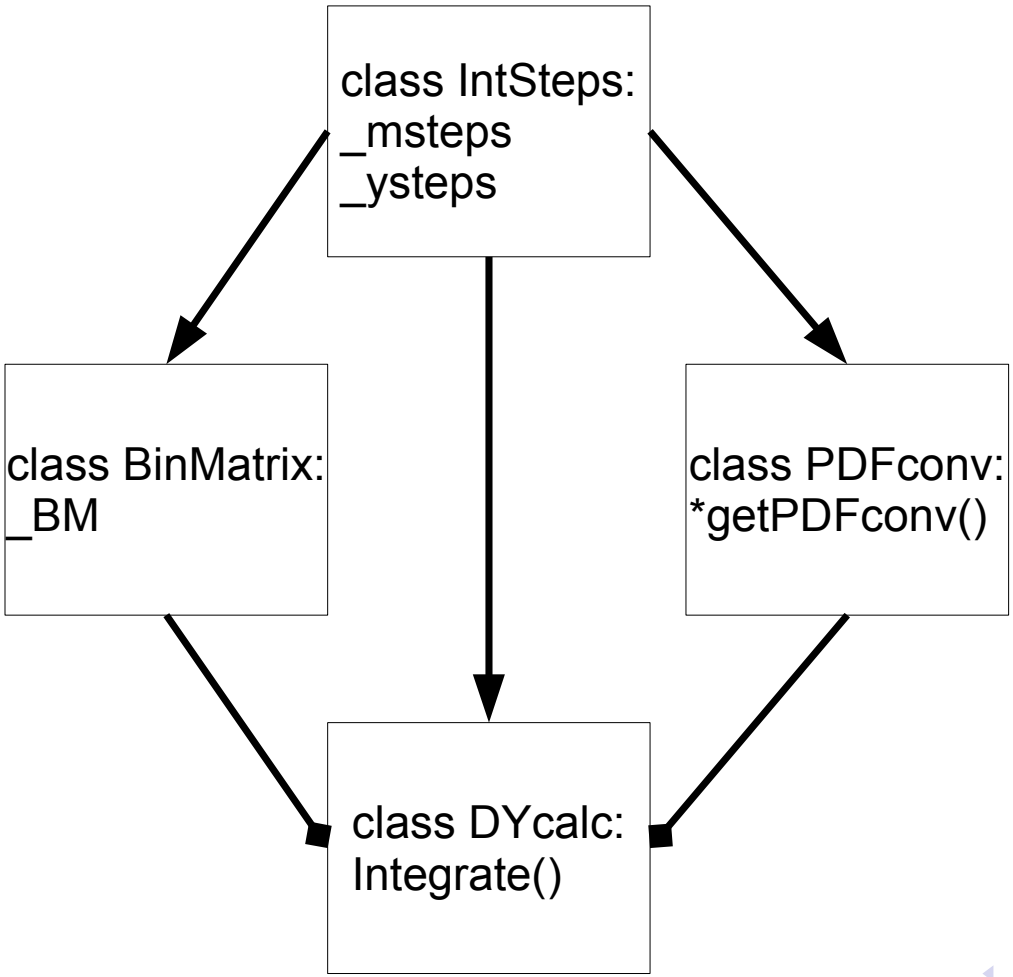
where $\tau = \frac{M^2}{S_0}$, S_0 - beam energy.
$F_{q\bar{q}}(1+\cos^2\theta^*, \cos\theta^*)$ is a linear homogenious dependence on $1+\cos^2\theta^*$ and $\cos\theta^*$.

- Integral over $\cos\theta^*$ is taken analitically in limits defined through eta ranges.
- $p_T(l)$ cut is possible
- dMdy needs to be integrated numerically since non-analylital PDF dependence.
- by Simpson method to avoid MC fluctuations
- numerical Simpson integration have customly optimized grid for dM and dy
- PDF part factorizes and is the only part that changes in the iteration.
- Makes simultaneous integration of several η_e bins possible
- Speed improvements by using precalculated non-PDF component of the expression in the Simpson integration steps.

QCDNUM optimizations

- QCDNUM allows fast structure function interpolation if grid points are fixed - in our case Simpson's steps.
- use indexed array of grid points (precalculated)
- use FASTSNS to initialize QCDNUM grid (standard limit on grid points in qcdnum.inc maybe not enough)
- perform interpolation and multiply
- class PDFconv

Class diagram



Code interface description:

Initialize:

```
int dy_create_calc_(const int *ds_id, const int *chg_prod,
    const double *beam_en, const char *bos,
    const double *ranges, const char *var_name,
    const int *n_bins, const double *bin_edges);
```

- `ds_id` - dataset identificator everything will be referenced by.
- `chg_prod, beam_en` - beam parameters.
- `bos` - boson name, "W" or "Z"
- `ranges` - 7 element array with m, y, eta ranges and pt cut.
- `var_name` - binned variable name, "eta" or "y"
- `n_bins, bin_edges` - binning parameters.

Calculate:

```
int dy_do_calc_();
int dy_get_res_(const int *ds_id, double *calc_res);
int dy_release_();
```

- `dy_do_calc` - performs calculations for all requested datasets.
- `dy_get_res` - writes calc results for dataset `ds_id` to array `calc_res`
- `dy_release` - frees allocated memory.

Fortran calls example:

- in InitDYCCXsectionDataset_kfactor subroutine in src/init_theory.f define input parameters: m(2), y(2), eta(2) ranges and pt_cut in ranges(7) array
- define beam energy and charge product and integration bins array eb(n_data_points)
- call calculation initialization routine for a given dataset ID IDataSet:

```
call dy_create_calc(IDataSet, bchpr, sqrtS, 'W'//char(0),
$     ranges, 'eta'//char(0), NDATAPOINTS(IDataSet), eb)
```

- The string constants define which bozon type is calculated for which distribution
- Most of the input parameters are read in automatically from datafiles

Fortran calls example:

- For each iteration do

```
call dy_do_calc;
```

- and read result (in src/dy_cc_sigma.f) to bsigs(n_data_points) array;

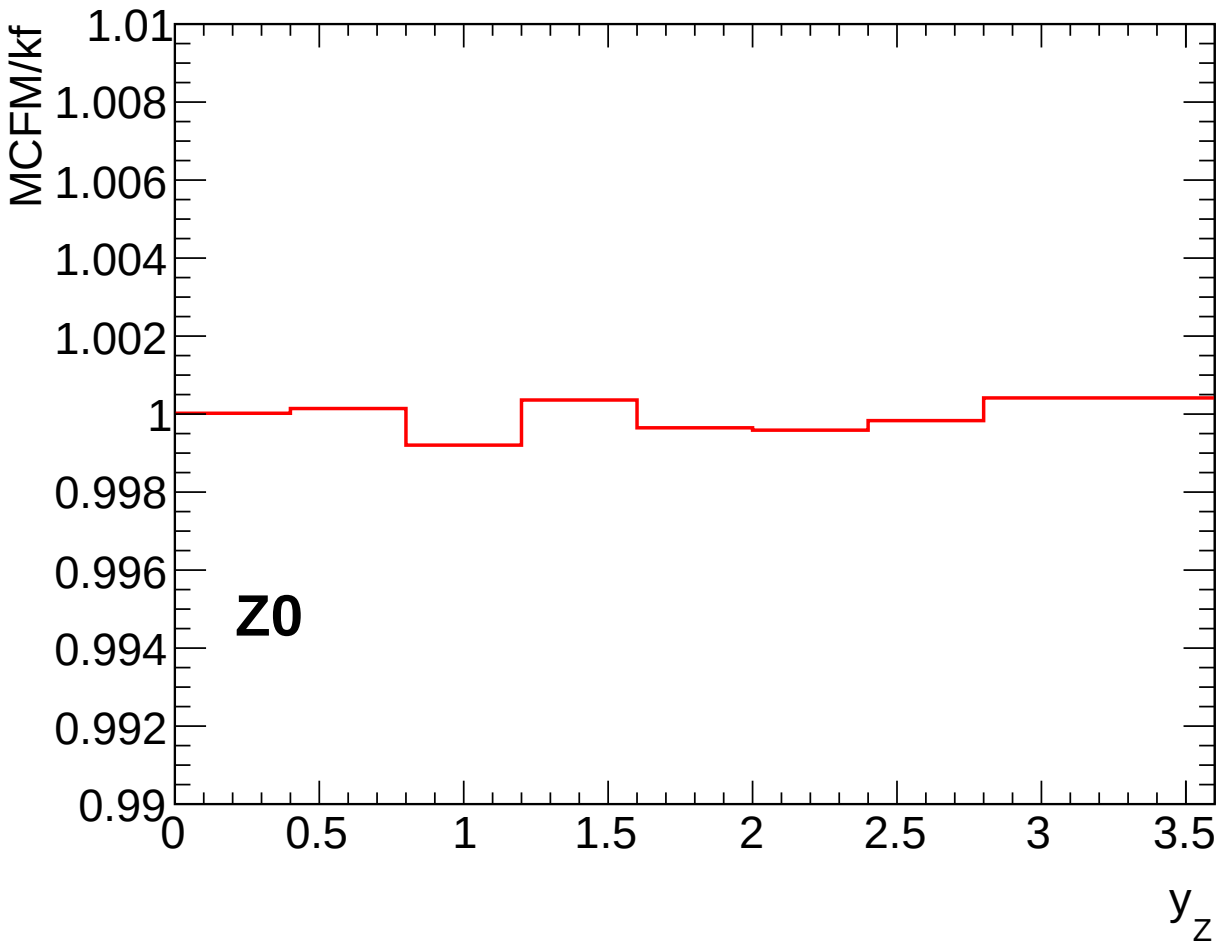
```
call dy_get_res(IDataSet, bsigs)
```

- when the run is finished, realese the memory by

```
call dy_release;
```

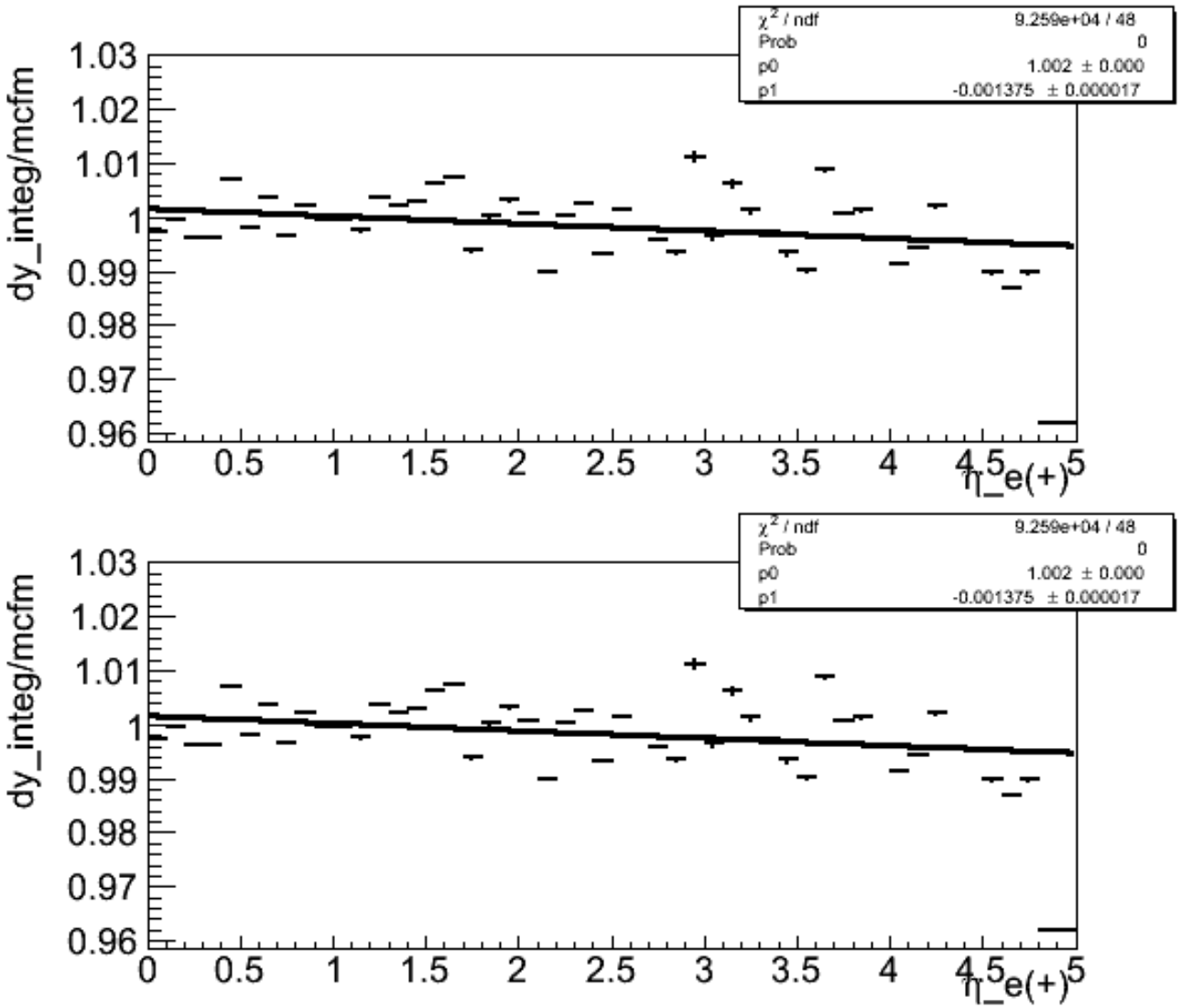
Validation with MCFM

HERAPDF1.5NLO via LHAPDF, MCFMv6.1



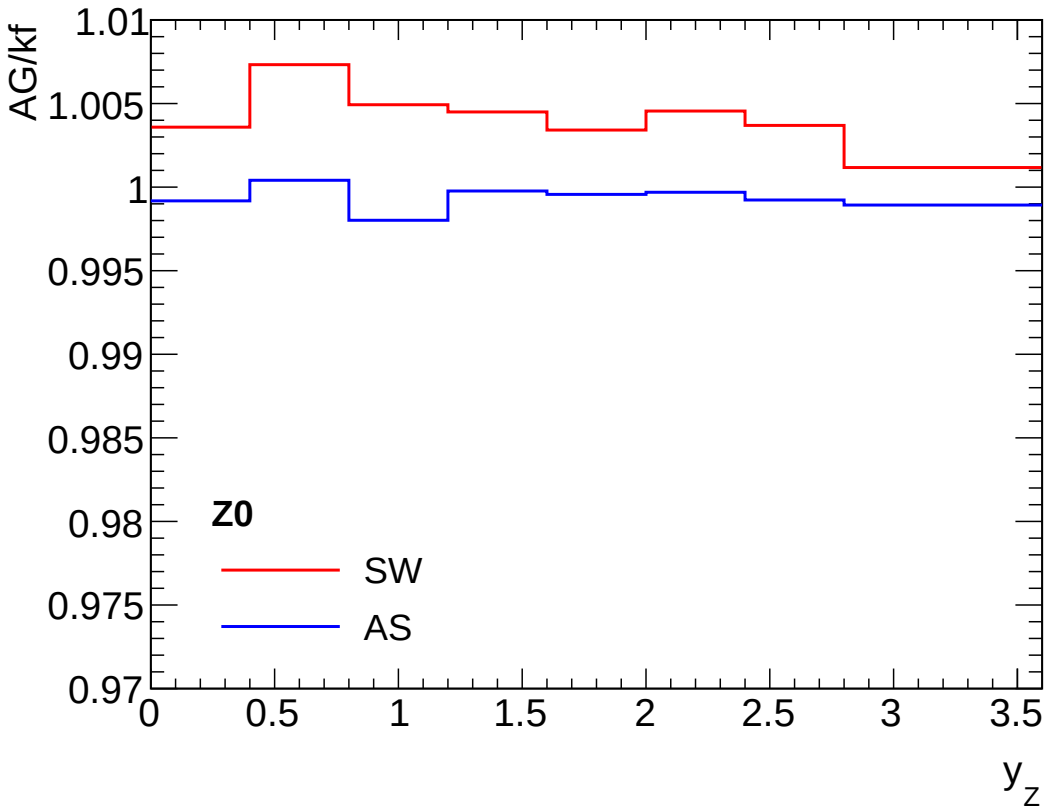
Validation with MCFM

HERAPDF1.5NLO via LHAPDF, MCFMv6.1



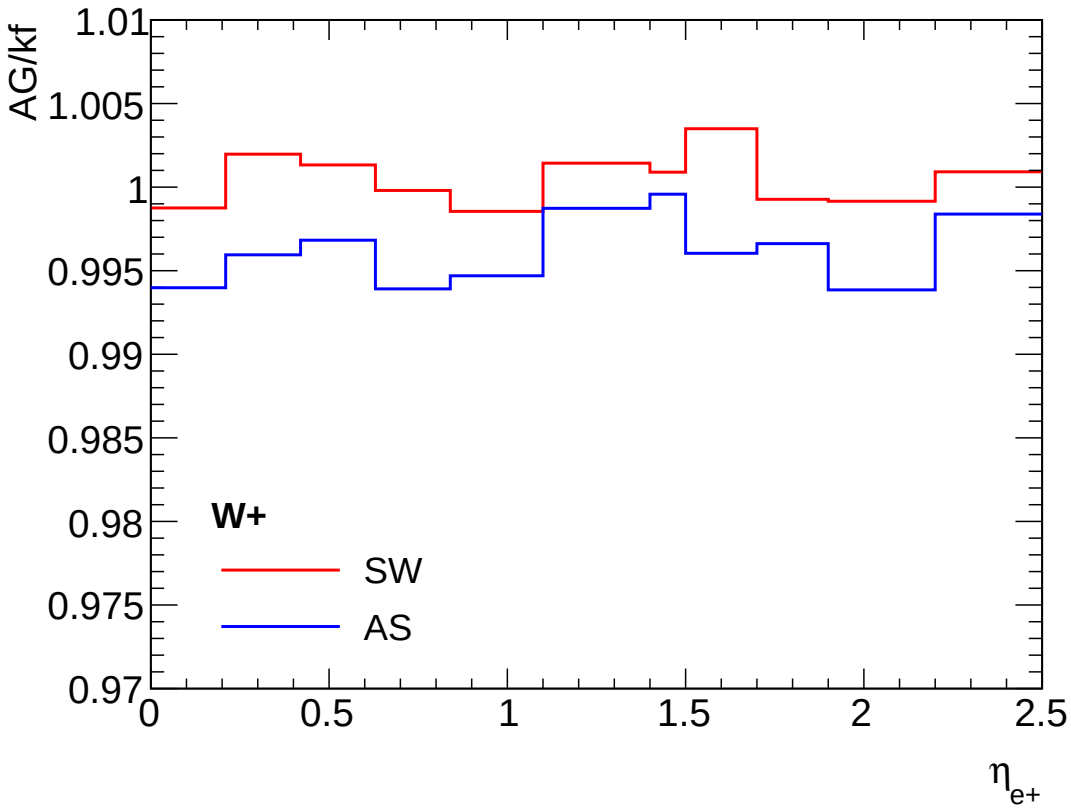
Crosscheck with APPLgrid

within HERAfitter, with APPLgrid 1.1.6, k-factors from MCFM6.1



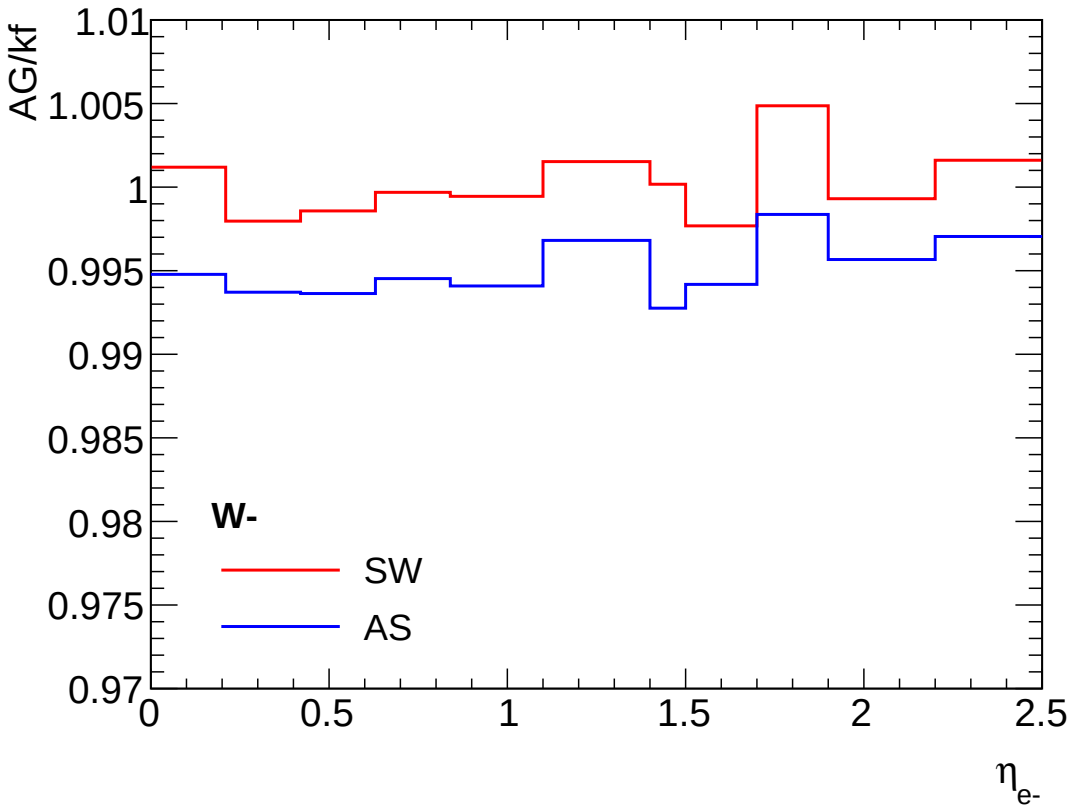
Crosscheck with APPLgrid

within HERAfitter, with APPLgrid 1.1.6, k-factors from MCFM6.1



Crosscheck with APPLgrid

within HERAfitter, with APPLgrid 1.1.6, k-factors from MCFM6.1



Summary

- The LO DY integration with NLO/NNLO k-factors is a fast tool in HERAfitter, which should allow to perform fits on it's own and make cross checks with APPLgrid.
- The independent comparison is quite difficult to conduct due to numerous parameters and conditions have to be complied.
- Cross checks with applgrid are ongoing
- Further steps would be to check how much the parameters variation in HERAfitter makes k-factor method deviate from APPLgrid results.