Code for dynamic-β simulations

A. A. Babaev for CMS BRIL project

Tomsk, 2017

1. Implemented algorithm

Nature of dynamic-beta and idea of dynamic-beta correction:

- The beam-beam interaction leads to distortion of bunch shape
- The beam-beam interaction effectively inserts a quadrupole magnet at the IP
- The dynamic-β variation with beam separation reflects the variation of the strength of that quad during the vdM scan (→ beam-beam tune shift)
- The dyn.-β correction is tantamount to 'turning back' the quad to a common (arbitrary) reference setting β^*_{ref} at each scan point

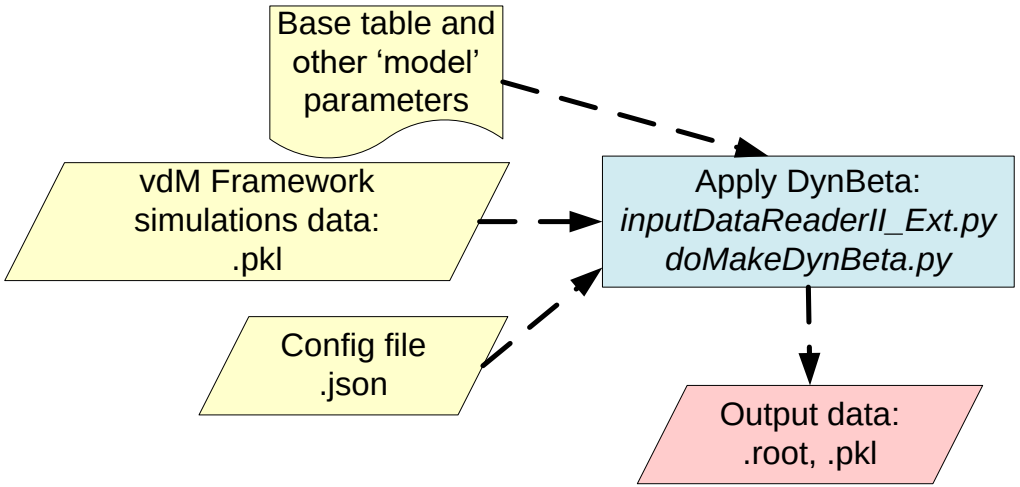
The code implements the prescription presented at:

- W. Kozanecki, Updated implementation of the dynamic-β correction, LLCMWG meeting, 5 Aug 2013
- A.A. Babaev, Dynamic-beta simulations for Fills 4266, 4954, BRIL DPG meeting, 30 May, 2017

The developed code uses simulations data from vdM Framework where BeamBeam correction is applied. The dynamic-beta correction 'turns this data back' to the conditions at head-on collision (reference settings β^*_{ref})

2. *doMakeDynBeta.py* script: general review

The script *doMakeDynBeta.py* applies the dynamic-β correction for all scans in the fill.



- Base tabe contains the information on dynamic-b at 'model' conditions; other 'model' parameters: bunch population and and emittances. Base table is the file: *dynBetaCrctnTable_v1.0_22May13.txt*

- vdM Framework simulations data: .pkl tables from vdM Framework run with BeamBeam correction: *Scan_%Fill%.pkl*, *Rates_%Luminometer%_%Fill%.pkl*, *BeamCurrents_%Fill%.pkl*, *BeamBeam_%Fill%.pkl*, *%Fit%_FitResults.pkl.*
- Config file contains pathes where .pkl files are placed and other parameters.
- *inputDataReaderII_Ext.py* is the class definition to store vdM data for single scan. It is based on the regular *inputDataReaderII.py* where several members and functions have been added.
- *doMakeDynBeta.py* - main script implementing the dynamic-beta correction algorithm.
- Output data: *graphs_%Fill%_DynBeta.pkl* and *graphs_%Fill%_DynBeta.root* prepared in the same format as usual vdM Framework graphs files (from *doMakeGraphFileII.py*).

3. *doMakeDynBeta.py* script: config file, output

Parameters in config file:

"InputScanFile": path to *Scan_%Fill%.pkl*
"InputBeamCurrentFile": path to *BeamCurrents_%Fill%.pkl*
"InputRatesFile": path to *Rates_%Luminometer%_%Fill%.pkl*
"InputBeamBeamFile": path to *BeamBeam_%Fill%.pkl*
"InputFitResultsFile": path to *%Fit%_FitResults.pkl* (after BeamBeam correction). This file is used to estimate the emittance.
"Scanpair": list of [ScanX,ScanY] scanpairs as at BeamBeam correction

doMakeDynBeta.py could be run by the usual way:
python doMakeDynBeta.py %path_to_config_file%

Output will be written in sub-directory *./graphs/*
If *./graphs/* does not exist, it will be created in the directory where *doMakeDynBeta.py* is placed.

4. Fit and visible cross-section calculation

As the output of *doMakeDynBeta.py* has the same format as the output of *doMakeGraphFileII.py* one can use usual scripts for fits and visible cross-section calculation without changes.

***To use a fit*:**

One should turn off all scripts in *vdmDriverII.py* (i.e. put 'false') excepting "runVdmFitter": true

After that in "vdmFitterConfig" section in the parameter "InputGraphsFile" one should point out the path to *graphs_%Fill%_DynBeta.pkl.* Run the *vdmDriverII.py.* Results will be in *<AnalysisDir>/<Luminometer>/results/<Corr>*. So, if one will point out, for example, "Corr": ["DynBeta"] in the config file, the separate folder *DynBeta* will be created in *<AnalysisDir>/<Luminometer>/results/.*

Results are the usual .pdf (file with fitted graphs), .scv and .pkl files (tables with fit results).

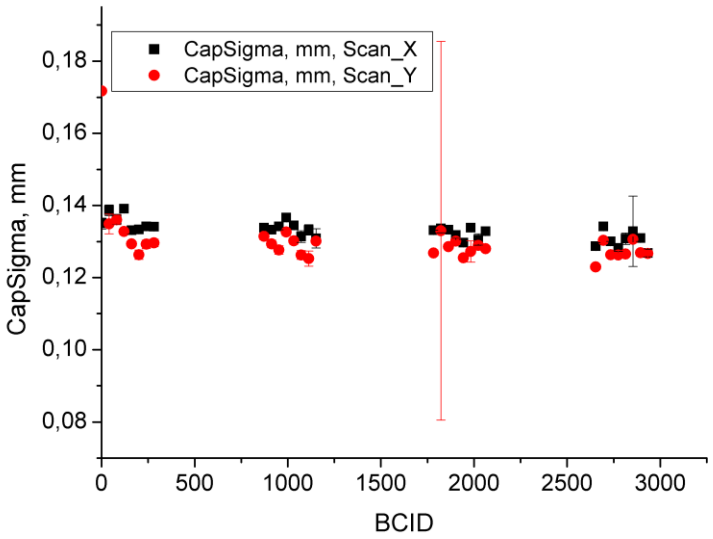
Or one can use *vdmFitterII.py* directly.

To calculate visible cross-section:

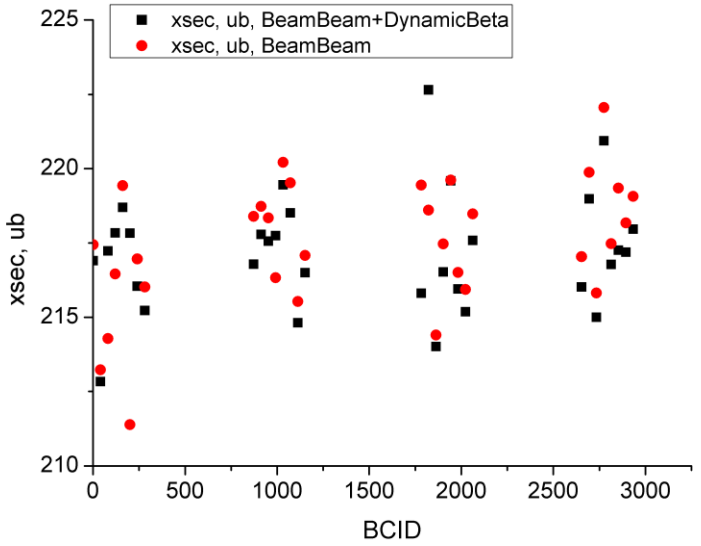
Run calculate *CalibrationConstant.py* with the appropriate path in "InputFitResultsFile" parameter of config file. Change "Corr" parameter to obtain the result in separate sub-directory.

5. Simulations for Fill 4954

Fill 4954: 27.05.2016, p+p; Energy: 6500 GeV; BetaStar: 1917 cm
Single X scan, single Y scan



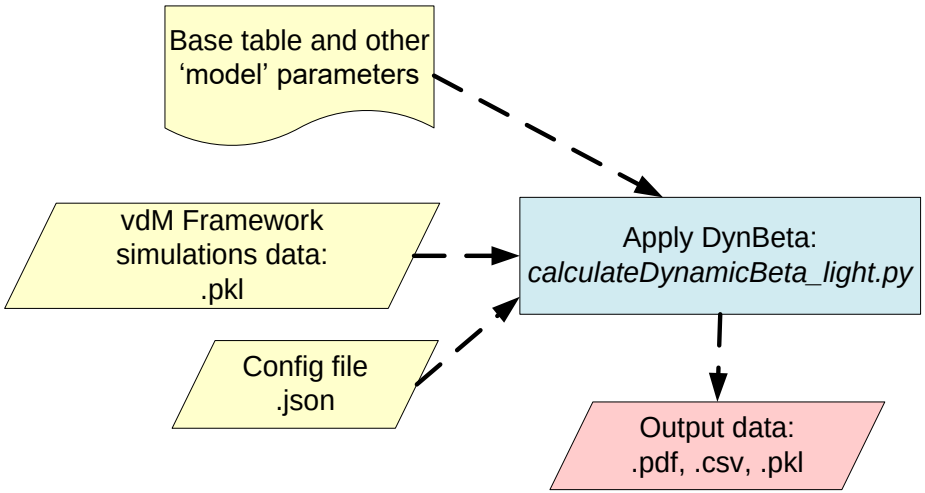
CapSigma vs BCID for both scans.
Luminometer - BCM1F.



Visual cross-section (xsec) vs BCID for two cases: with and without of Dynamic-β. As a rule, xsec when Dynamic-β is taken into account is less than the xsec at BeamBeam correction only. The typical difference is about 1-3 ub (i.e about 1%)

6. *calculateDynamicBeta_light.py*: introduction

The script *calculateDynamicBeta_light.py* applies the dynamic-β correction for single BCID of colliding bunches at one scanpair. It was designed to visualize correction curves and luminosity correction factors (see in A.A. Babaev, ... BRIL DPG meeting, 30 May, 2017, slides 2.3, 2.6).



- Base tabe contains the information on dynamic-b at 'model' conditions; other 'model' parameters: bunch population and and emittances. Base table is the file: *dynBetaCrctnTable_v1.0_22May13.txt*

- vdM Framework simulations data: .pkl tables from vdM Framework run with BeamBeam correction.
- Config file contains pathes where .pkl files are placed and other parameters.
- *calculateDynamicBeta_light.py* - main script. It finds in .pkl information for the pointed BCID and applies the dynamic-beta correction.
- Output data: *DynamicBeta_light_%Fill%_%BCID%_%ScanX%_%ScanY%.pkl, .csv, .pdf* - is written in the sub-directory *DynBeta_light_results*

7. *calculateDynamicBeta_light.py*: config file

Parameters in config file:

"ScanFilePath": path to *Scan_%Fill%.pkl*
"CurrentFilePath": path to *BeamCurrents_%Fill%.pkl*
"RateFilePath": path to *Rates_%Luminometer%_%Fill%.pkl*
"BeamBeamFilePath": path to *BeamBeam_%Fill%.pkl*
"FitResultsFilePath": path to *%Fit%_FitResults.pkl* (after BeamBeam correction). This file is used to estimate the emittance.
"BCID": BCID considered
"Scanpair": [ScanX,ScanY] scanpair; ScanX - first position, ScanY - second position
"FitName": fit name to be applied for CapSigma evaluation
"FitConfigFile": path to the fit config file

calculateDynamicBeta_light.py could be run by the usual way:
python calculateDynamicBeta_light.py %path_to_config_file%

8. *calculateDynamicBeta_light.py*: output

DynamicBeta_light_%Fill%_%BCID%_%Scan_X%_%Scan_Y%.pdf
The file contains three plots: the correction curves, the luminosity correction factor for X scan, the luminosity correction factor for Y scan.

DynamicBeta_light_%Fill%_%BCID%_%Scan_X%_%Scan_Y%.csv
The table contains the next information:
- Fill, Energy[GeV], BetaStar[m], numbers of Scan_X and Scan_Y, BCID, normalized emittances for X and for Y scans [um*rad];
- Fit results after the correction (the same results for all BCIDs in the fill could be obtained as described in slide 4);
- Numerical data for correction curves;
- Numerical data for luminosity correction factor.

DynamicBeta_light_%Fill%_%BCID%_%Scan_X%_%Scan_Y%.pkl
The table contains the binary copy of .csv data. The key to get numerical data for correction curves is “DBTable”. The keys to get numerical data for luminosity correction curves are “LCFTableX" (for X scan) and “LCFTableY" (for Y scan).

9. Concluding remarks

1. By default the presented code is placed in separate sub-directory *DynBetaScript* under vdM framework directory. *DynBetaScript* contains files *doMakeDynBeta.py, inputDataReaderII_Ext.py, calculateDynamicBeta_light.py, dynBetaCrctnTable_v1.0_22May13.txt* and subdirectories *graphs* and *DynBeta_light_results*. Config files could be placed anywhere.

2. The base table allows the DynBeta correction within the range of beam separation between -6.9Σ and 6.9Σ. If scanpoint is out of this range the scanpoint is excluded, the Warning is generated. This is normal.

3. There is the possibility to include this code in vdmDriverII.py as separate section. Also, one can make additional work to include the DynBeta in the list of available corrections and then to treat it as the regular correction.