Summary of the meeting on vertexing

FCC physics meeting
Feb 22, 2021
C Helsens
CERN-EP

Topical meeting on vertexing, Feb 10, 2021:

- organised jointly by Physics Performance & Physics Software and Computing
- https://indico.cern.ch/event/1003610

Goal of the meeting

Set of software tools to serve the needs of the FCC community

- Vertexing often came up in our Physics Performance meetings
  - Crucial tool for many analyses
- Last time we discussed about vertexing “in practice” was in October
  - https://indico.cern.ch/event/965346/
  - First attempt to run the ILC algorithm over FCC event files
  - Cumbersome, had to convert to LCIO (ILC format) first, …
- Quite some progress since then, so now is a good time to go through
  - Activities that developed recently
  - New developments that could start now/soon
  - The integration of (some) existing vertexing software

Goal of the meeting

In the process of defining requirements on the vertex detector: we should make use of the best possible vertexing tools in the sensitivity analyses.

The meeting goal was to help us review the current status and define projects to move forward in this direction.

Attendance & minutes of the discussions

Special FCC P&P Software Meeting, 10 Feb 2021

Present: None; Vidyo only meeting following CERN coronavirus restrictions
Remote: E Perez, P Azzi, G Ganis, C Helsens, V Volkl, A Salzburger, F Bedeschi, A Akhundov, Ang Li, Aridam Das, F Brieuc, D Contardo, Hwidong Yoo, I Korzhavina, J Alcaraz, L Gouskos, L Poggioli, M Boscolo, M Chrzaszcz, P Gesinger-Befurt, P Fernandez, R Sengupta, R Aleksan, S Heinmeyer, W Hulsbergen, Ziad El Bitar, M Selvaggi, F Grancagnolo, D Hill, B Schlag, A Blondel, G Wilkinson, M Dam, S Monteil, BFL Ward, E Gorini, J Smiesko, G Tassielli, P Azzurri, V Diolaiti, Sanghyun Ko, J Andrea, K Gautam, A Sailer
Agenda: https://indico.cern.ch/event/1003610/

The following are some notes taken during the discussions which followed each talk. Please refer to the slides for the content of the talks themselves.

Minutes of the discussions were taken and are linked to the Indico agenda.

Agenda

https://indico.cern.ch/event/1003610

FUTURE CIRCULAR COLLIDER

FCC-ee topical meeting on vertexing - combined Phys Performance + Software

Wednesday 10 Feb 2021, 14:00 → 17:00 Europe/Zurich

Clement Helsens (CERN), Emmanuel Francois Perez (CERN), Gerardo Ganis (CERN), Patrizia Azzi (INFN Padova (IT))

14:05 → 14:15 **Physics motivations**
Speaker: Roy Aleksan (Université Paris-Saclay (FR))

14:15 → 14:35 **A stand-alone vertex-fitting algorithm**
Speaker: Franco Bedeschi (Universita & INFN Pisa (IT))

14:35 → 14:55 **First example resolutions of displaced vertices in exclusive processes (using FB's algo)**
Speakers: Clement Helsens (CERN), Emmanuel Francois Perez (CERN)

14:55 → 15:25 **The DecayTreeFitter algorithm**
Speaker: Wouter Hulsbergen (Nikhef National institute for subatomic physics (NL))

15:25 → 15:45 **Implementation of the LCFI+ algorithm into key4hep**
Speakers: Andre Sailer (CERN), Placido Fernandez Declara (CERN)

15:45 → 16:05 **Status of ACTS (tracking and) vertexing, and implementation in key4hep (t.b.c.)**
Speakers: Andreas Salzburger (CERN), Bastian Schlag (CERN / JGU Mainz), Paul Gessinger-Befurt (CERN / JGU Mainz)

Follow up - 1: FCC-ee P&P

https://indico.cern.ch/event/1007009/

FCC-ee Physics Performance meeting -

Monday 15 Feb 2021, 15:00 → 17:00 Europe/Zurich

Emmanuel Francois Perez (CERN), Patrizia Azzi (INFN Padova (IT))

Description CONNECTION WILL BE USING ZOOM. INSTRUCTIONS BELOW

NO PHYSICAL ROOM

zoom.txt

15:50 → 16:05 **Summary of the meeting on vertexing** 15m

Speakers: Clement Helsens (CERN), Emmanuel Francois Perez (CERN), Gerardo Ganis (CERN), Patrizia Azzi (INFN Padova (IT))

2021_02_15_summ...

Follow up - 2: LLP @FCC-ee https://indico.cern.ch/event/1001669/

Try to engage the LLP group in vertexing studies using common SW tools

Follow up - 3: today's meeting

1. Detailed summary of the meeting

2. Progress since the meeting

3. Summary

1. Detailed Summary of the meeting

Physics motivations - 1

Vertexing : An indispensable tool for precision Physics

R. Aleksan

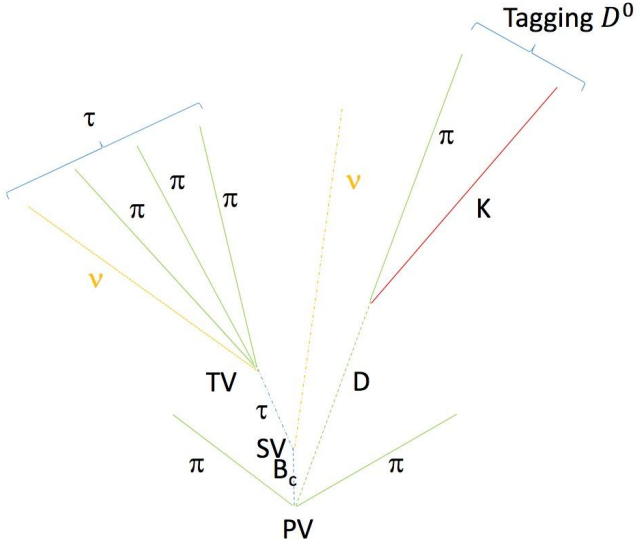
- Time dependent measurements (e.g. CP violation studies…)
  - B flight distance measurement
  - B-Tagging
- Electroweak and Higgs Physics
  - b-tagging, c-tagging, τ-tagging
- Rare decays
  - Limiting the combinatorial background
  - Reconstruction of final states with neutrinos
- ….

Physics motivations - 2

R. Aleksan

Rare decays

$B_c^{\pm} \to \tau^{\pm}\nu$



$Br(B_c^{\pm} \to \tau^{\pm}\nu) \approx 5\%$

$\frac{N(B_c^{\pm})}{N(B_u^{\pm})} \leq 10^{-3}$

$\Rightarrow \approx 4 \cdot 10^7 \; B_c^{\pm} \to \tau^{\pm}\nu$

$< p_\tau > \approx 19 \; GeV, < d_\tau > \approx 0.9mm$

Two missing neutrinos and no information on SV but some interesting features to reduce background:

- Tagging b in opposite side
- Same side tagging with D
- B_c flight very short ($c\tau \approx 150\mu m$) compared to $B^+ \to \overline{D^0}\tau^+\nu$ or $B^0 \to D^-\tau^+\nu$ (very dangerous background (> x 10^2)

Physics motivations - 3

Conclusion

R. Aleksan

Vertexing is a vital tools in variaty of precision measurements @ FCC

Let’s get to work

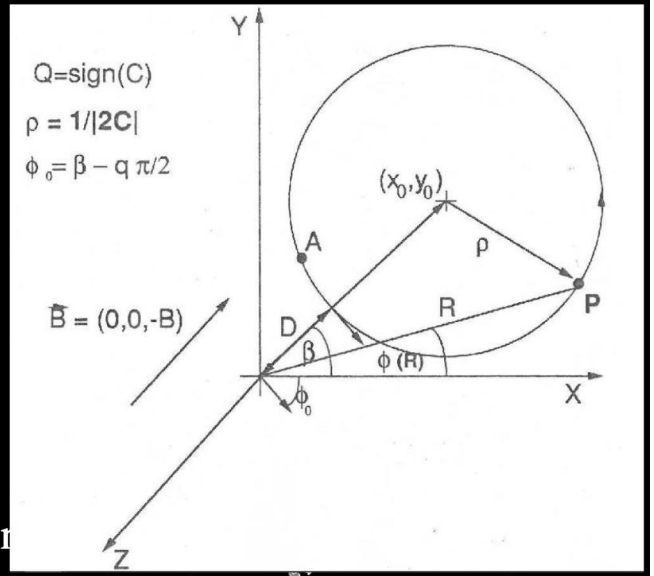
➜ So we did!

A stand-alone vertex-fitting algorithm - 1

F. Bedeschi

Vertex fit (1)

- **Input:**
  - Array of observed tracks
    - 5 parameters + Cov. Matrix
    - Assume perfect helix
      - D, ϕ_0, C, cot θ, z_0
- **Output:**
  - 3D vertex + covariance
- **Method:**
  - χ^2 minimization with constrai
    - Vary parameters and phase and force all tracks to cross at same point

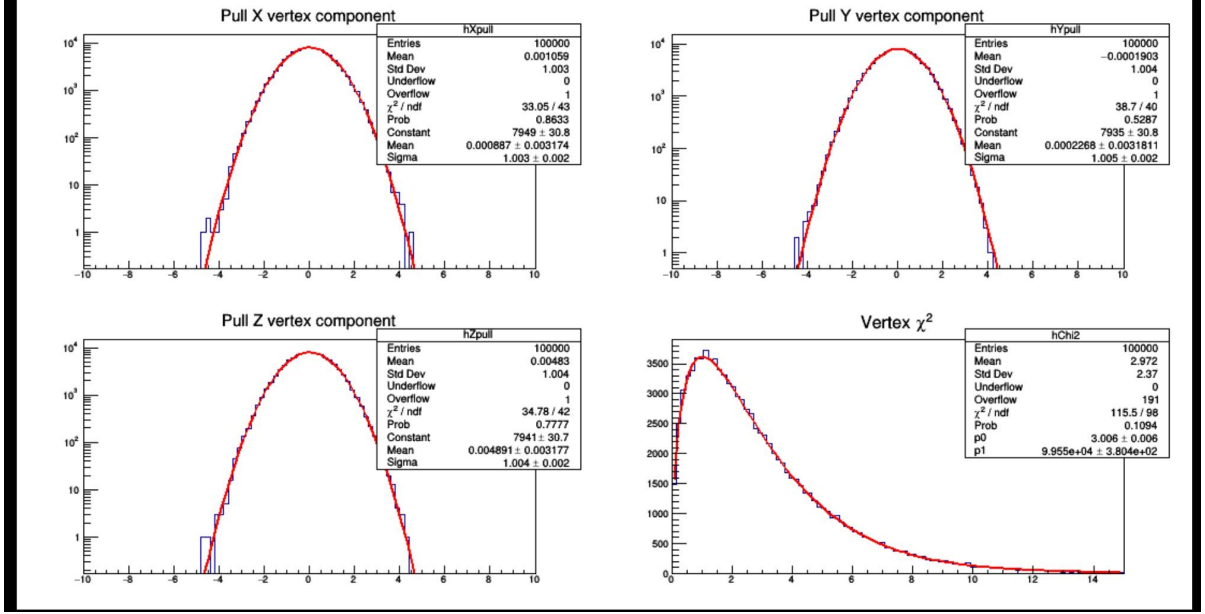


A stand-alone vertex-fitting algorithm - 2

F. Bedeschi

Performance (1)

- Example: 100,000 events, 3 tracks, R_V (0-1.5 cm), pt (0.5-1.0 GeV)



A stand-alone vertex-fitting algorithm - 3

F. Bedeschi

Way forward

- Can turn routine into a class with additional features:
  - External vertex constraint (easy): useful for primary vertex
  - Adding/removing individual tracks (easy)
  - Calculate updated track parameters and their error matrix
    - Large matrix with correlations (easy)
  - Setup for tertiary vertices (more complex)
    - Calculate total vertex momentum /covariance
    - Allow for vertices with neutral tracks
- In general strong interaction with track class format
- Simple track/vertex structure to be refined

First example resolutions of displaced vertices in exclusive processes (using FB's algo) - 1

E. Perez; CH

Setup

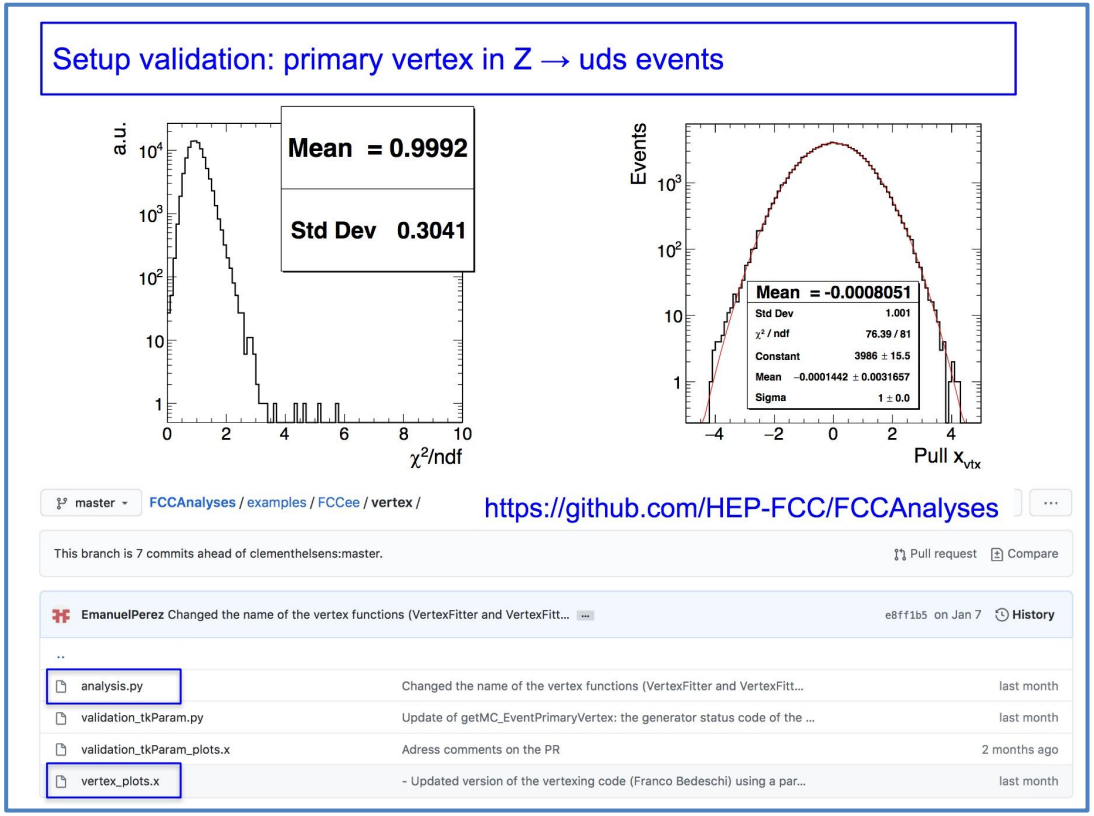
- Delphes samples in the EDM4HEP format
- Z → bb events at 91 GeV generated with PYTHIA + EVTGEN. Exclusive decays studied :
  - Bs → J/ψ Φ → μμKK,
  - Bc or B+ → τ ν → 3π $\nu\nu$,
  - Bs → Ds K → (Φ π) K → (K K π) K
- Delphes uses the IDEA card, with the TrackCovariance module to produce 5-parameters tracks
- Analysis within the FCCAnalyses framework
  - VertexFitter code from FB implemented
  - Actually not the latest version from his code

Caveats:
Only the diagonal elements of the covariance matrix of the track parameters are stored currently in the EDM4HEP files. May affect a bit the vertex fits shown here.

- does not seem to have a large effect when fitting a primary vertex with many tracks. But may have a larger effect here.

First example resolutions of displaced vertices in exclusive processes (using FB's algo) - 2

E. Perez; CH



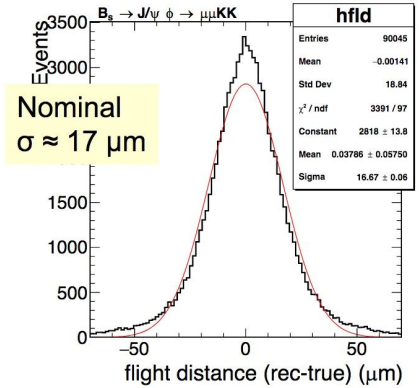
➜ Working and documented examples already available at the time of the meeting

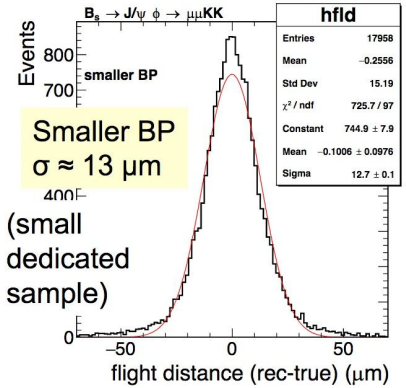
First example resolutions of displaced vertices in exclusive processes (using FB's algo) - 3

E. Perez; CH

Bs → J/ψ Φ → μμKK: beam-pipe radius

- All what was shown previously corresponds to the “nominal” beam-pipe with an inner radius of 1.5 cm : first layer of the VTX sits at 1.7 cm from the beam-line.
- Under consideration: smaller BP, R = 1 cm → VXD at 1.2 cm





A setup is in place that allows to study: different radiii, different single-point resolution, the effect of the thickness of the layers / beam-pipe, etc.

Detailed studies can be started right now !
Volunteer ? Contact us !

➡ people have already contacted us.

First example resolutions of displaced vertices in exclusive processes (using FB's algo) - 4

E. Perez; CH

Conclusions

The vertex fitter of Franco allows several interesting studies to be made

- a determination of the expected resolutions on displaced vertices in chosen examples, as shown here
  - Very first results presented here, to be consolidated & pursued
  - The analysis shown here will be put in the central repository and documented
- An estimation of the effect of variations of the detector model: change the radii of the layers of the vertex detector, the thickness of the layers or the BP, the single-hit resolution
  - This can be pursued already now with the tools that are in place !
- Other short term projects: e.g. write a vertex-finding algorithm that runs the fitter iteratively, to determine first the primary vertex, and then displaced vertices, in order to tag “b-jets” inclusively.
  - Could be started now.

And other projects within, or in conjunction with, the other algorithms that will be described in the next talks.

The DecayTreeFitter algorithm - 1

W. Hulsbergen

what is it?

- "Decay Tree Fitter" (a.k.a. "DTF")
  - least squares algorithm
  - extracts four-momenta, decay times, vertex positions from a decay chain
- developed for BaBar: now in use in LHCb, Panda, Belle-2, …
- original implementation is in C++
  - the LHCb code is basically just a fork of the BaBar code
  - Belle-2 code is independent (but inspired by the LHCb/BaBar code)
- code is not in a fantastic state:
  - >16 years old, lot's of dynamic allocation, still uses CLHEP!
  - happy to share it, but one could also start from scratch

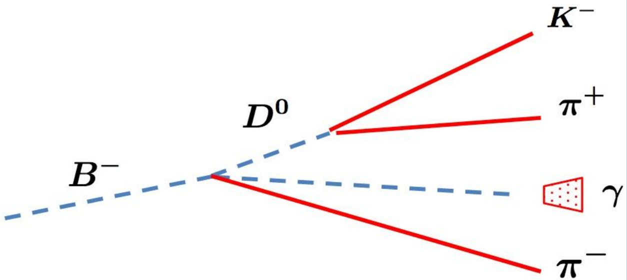
The DecayTreeFitter algorithm - 2

W. Hulsbergen

reconstructing a decay chain

- consider a multi-level decay chain

$$B^- \to D^{*0}\pi^-$$
$$\hookrightarrow D^0\gamma$$
$$\hookrightarrow K^-\pi^+$$



- traditional method: "leaf-by-leaf fitting"
  - fit most downstream vertices first
  - use composites as input to next upstream level
  - very natural way to reconstruct and select cascade decays

- to implement this, need to extend track-based vertex fit with constraints for
  - photons, merged pi0 (calorimeter clusters)
  - short-lived composites (e.g. D*, J/psi)
  - long-lived composites (e.g. Ks, D0, B+)

The DecayTreeFitter algorithm - 3

decay tree fitter for LHC-ee?

- it seems useful for the flavour physics program
  - LHCb/B-factory physicists will certainly appreciate it
- migration from BaBar to LHCb was very straightforward
  - adapt to different implementation of ‘particle’
  - adapt to different track/cluster models
- these things are well isolated in the code, so, it should be reasonably easy to do this for FCC-ee
- that said … the core needs real work too, for instance:
  - CLHEP → Eigen?
  - virtual inheritance & dynamic allocation → templates, variants, …
  - remove historical parts, like obsolete ordering of constraints

W. Hulsbergen

The DecayTreeFitter algorithm - 4

Conclusion

- Decay Tree Fitter is an implementation of a global decay chain fit
- used in several flavour physics experiments
- code base is C++
  - not experiment independent, but perhaps reasonably easy to adapt
  - may be a good student/postdoc project: a few months should be more than sufficient to (re)implement it

W. Hulsbergen

Implementation of the LCFI+ in Key4Hep - 1

P. Fernandez
A. Sailer

k4MarlinWrapper

- Part of the Key4hep project: https://github.com/key4hep/
- *Marlin* Processors functionality made available in Key4hep through the *Gaudi* framework.
- It contains the necessary interfaces to deal with *Marlin* formats to be run from *Gaudi* algorithms.
  - Wrapper around Marlin Processors
  - XML steering file to Python options file converter
  - EDM4hep to LCIO event converter in memory
- Marlin source code is kept intact, and can be called on demand.

Implementation of the LCFI+ in Key4Hep - 2

Conclusions

- The converter tool is able to run the "VertexFinder" LCFI+ Processor, converting the EDM4hep collections to LCIO *on the fly*.
- Conversion of the output collection back to EDM4hep needs to be integrated/implemented.
- More collection types to be converted are being supported.
- Feedback from real world usage appreciated.

➔ Feedback is being provided

P. Fernandez
A. Sailer

Acts - 1

project - community, friend & family

- ACTS as one project in a new ecosystem of community driven SW
  - We **walk** and **we learn** together
    - interaction between SW projects **under the umbrella of HSF** is key
    - Example: ACTS **report** on Eigen compilation restrictions/issues in HSF WG#2
      similar issues seen by other Eigen clients (CMS), follow up by HSF
  - We should **play** together
    - Encouragement to put modules together, build systems
    - Example: can ACTS run within on top of PataTrack, within ALLEN, etc…
  - Finally, we should **work** together

A. Salzburger
B. Schlag

Acts - 2

Current Vertexing Developments in ACTS

A. Salzburger
B. Schlag

Generalization of track linearization using the ACTS::Propagator:

$$\vec{q} = \vec{q}(\vec{r}, \vec{p}) = A\vec{r} + B\vec{p} + \vec{c}_0$$

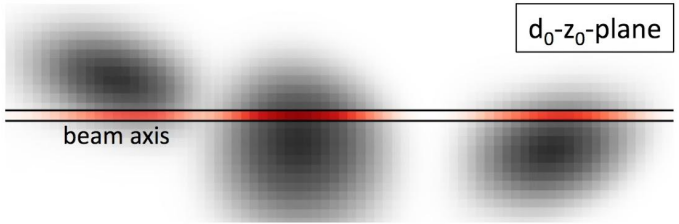
Retrieve dedicated Jacobians from ACTS::Propagator

- No assumption of helical track parameters anymore
- Vertex fitter more robust in all detector regions
- Harmonize primary and secondary vertexing with common math kernels
- Fully integrated time propagation in ACTS Vertex fitting with time information possible
- → Currently WIP and contributions are always welcome

Gaussian Grid Track Density Vertex Seed Finder:

- Model track as 2-dim Gaussian density grid in d_0-z_0-plane
  → calculate only track contribution along beam axis (red)
- Superimpose all tracks and find maximum along beam axis

Extremely fast in iterative approaches



Example: Track density representations of 3 single tracks

Acts - 3

Summary & Outlook

- **Modern, fast & MT-capable** vertexing suite implemented in ACTS
- Easy to integrate in any reconstruction framework

  → fully integrated in ATLAS reconstruction framework and set as default primary vertexing tool
- **Validated** in **single-threaded** and **multi-threaded** execution mode
- **2-3x faster** than original implementations
- **Generalization of track linearization** work ongoing
  - Harmonize vertex fitting for primary and secondary vertexing + include time information
  - Any contribution is very welcome → let us know if you are interested
- **New seed finder** available: Excellent physics & CPU performance

A. Salzburger
B. Schlag

2. Progress Since the Meeting

Vertexing algorithm situation as of last week

	Analysis level	Reco level	Speed	Neutrals	Implementation
Franco's fitter	✓	(✓)	✓	✗	✓
LCFI+	✗	✓	✗	✗	In progress
DecayTreeFitter	✓	✗	✗✗	✓	Contact established
ACTS	✓	✓	✓	✗	Proof of concept
LHCb PV algo.	✓	✓	?	✗	Started

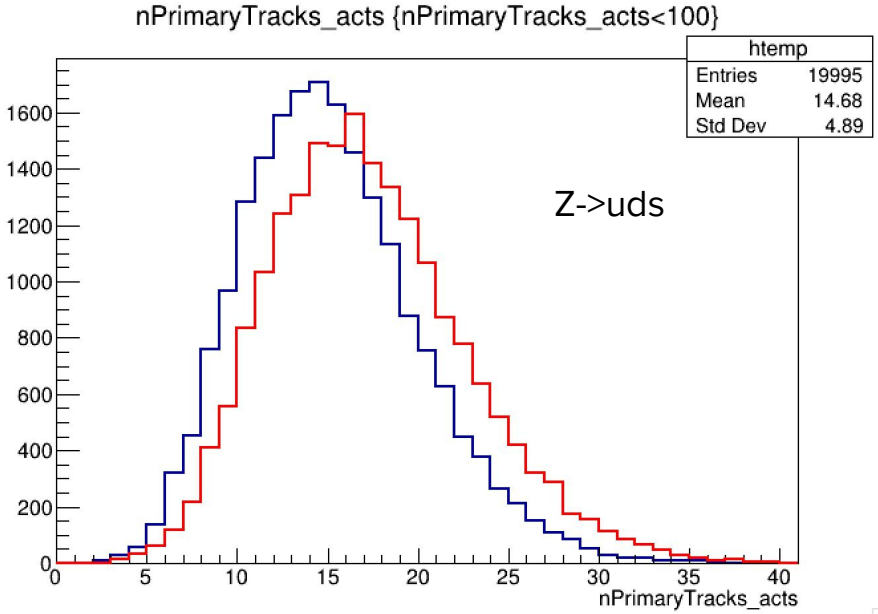
Vertexing algorithm situation as of now

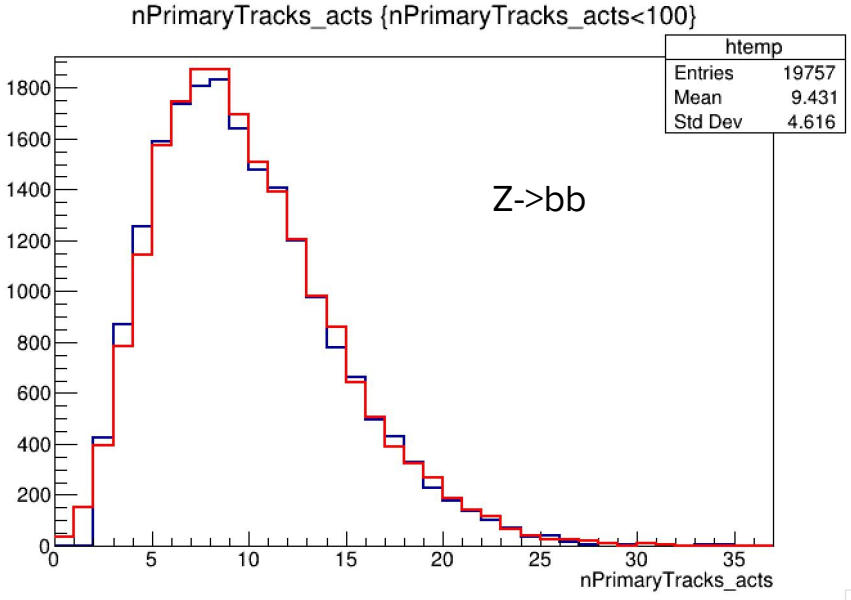
	Analysis level	Reco level	Speed	Neutrals	Implementation	Analysis examples
Franco's fitter	✓	(✓)	✓	✗	✓	✓
LCFI+	✗	✓	✗	✗	Close to completion	✗
DecayTreeFitter	✓	✗	✗✗	✓	Establishing a plan	✗
ACTS	✓	✓	✓	✗	Almost finalised at analysis level	✓
LHCb PV algo.	✓	✓	?	✗	Started	✓

Acts integration

- Acts AMVF vertex finding algorithm has been implemented in FCCAnalyses

Blue: Acts vertex finding (all tracks)
Red: FB vertex fitting using tracks associated to PV

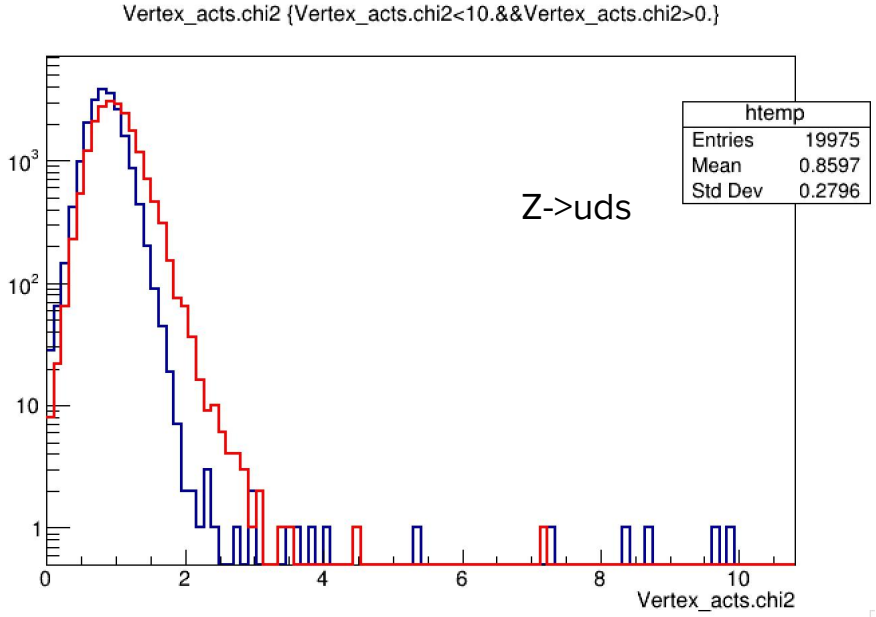


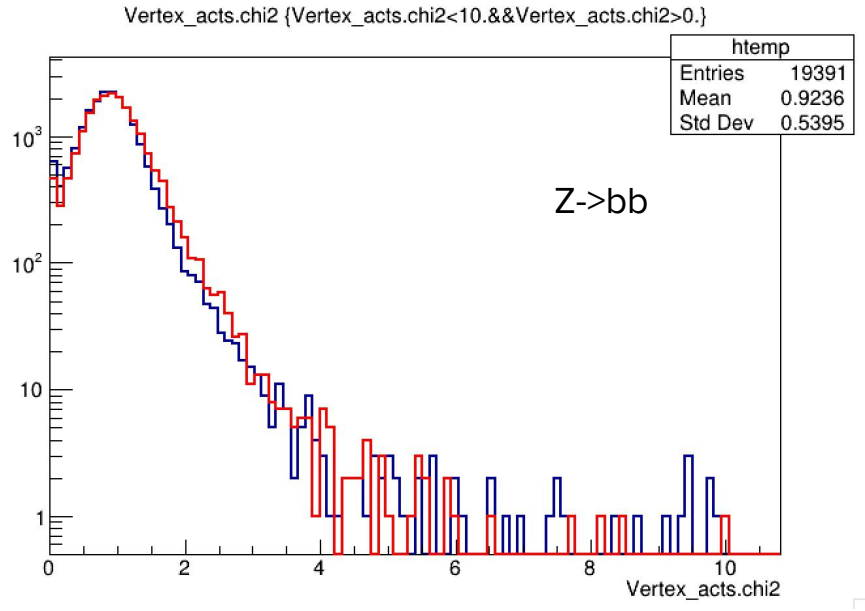


Acts integration

- Acts AMVF vertex finding algorithm has been implemented in FCCAnalyses

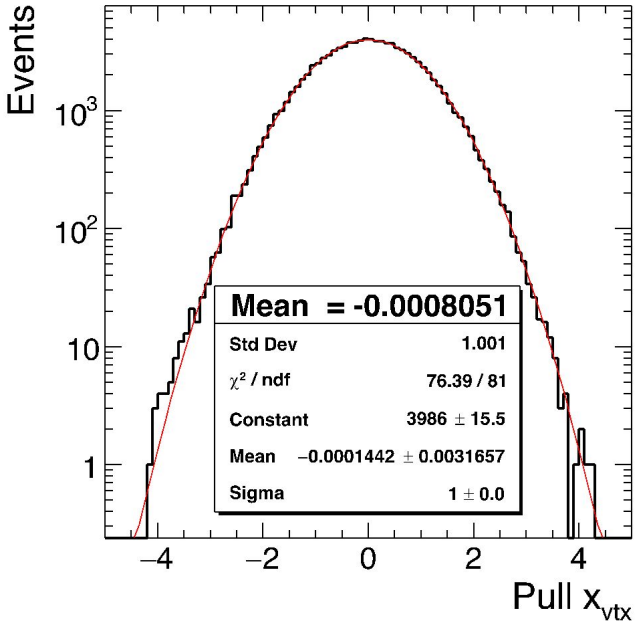
Blue: Acts vertex finding (all tracks)
Red: FB vertex fitting using tracks associated to PV

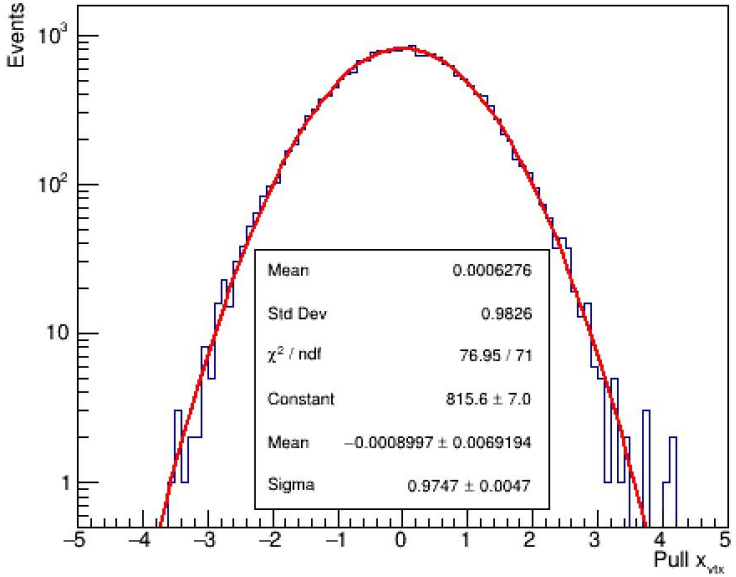




Acts integration

- Acts AMVF vertex finding algorithm has been implemented in FCCAnalyses





Acts integration

Very preliminary results need careful validation
But Acts can be used as primary vertex finder right now in FCCAnalyses.
Then FB vertex fitter (or Acts vertex fitter once implemented) can be used as we did already for displaced vertices.
Logic for DV finding still needs to be written though.

Off diagonal terms in covariance matrix - 1

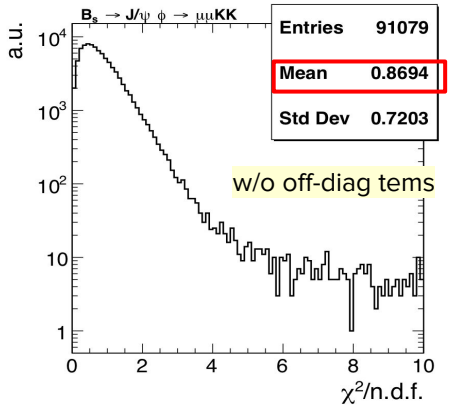
At the meeting of Feb 10:

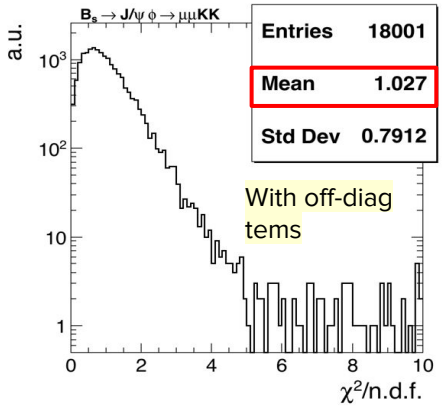
- Vertexing performances on exclusive B decays, obtained in FCCAnalyses using the vertex fitter of Franco B, had a caveat:
  - Only the diagonal elements of the covariance matrix of the track parameters were stored in the files.
  - Was the suspected reason for non-perfect chi2 and pulls of the fits.

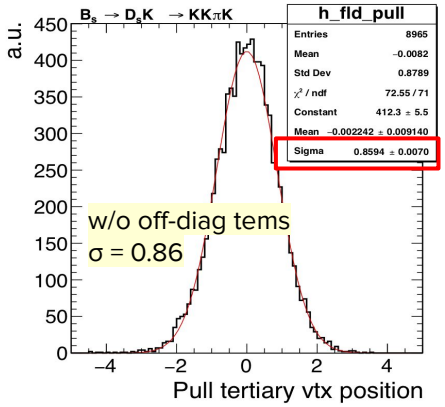
Off diagonal terms in covariance matrix - 2

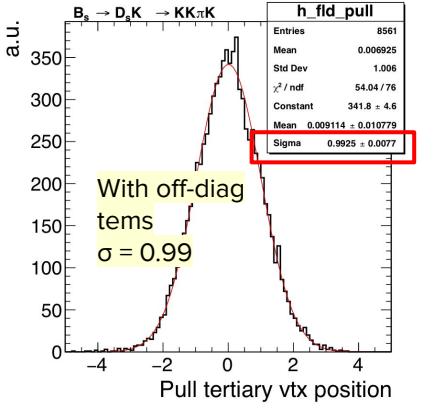
As of now

- The off-diagonal terms are now available (updates to DELPHES and to k4SimDelphes).
- The quality of the fits is now much better ! Examples (vtx resolutions largely unchanged)



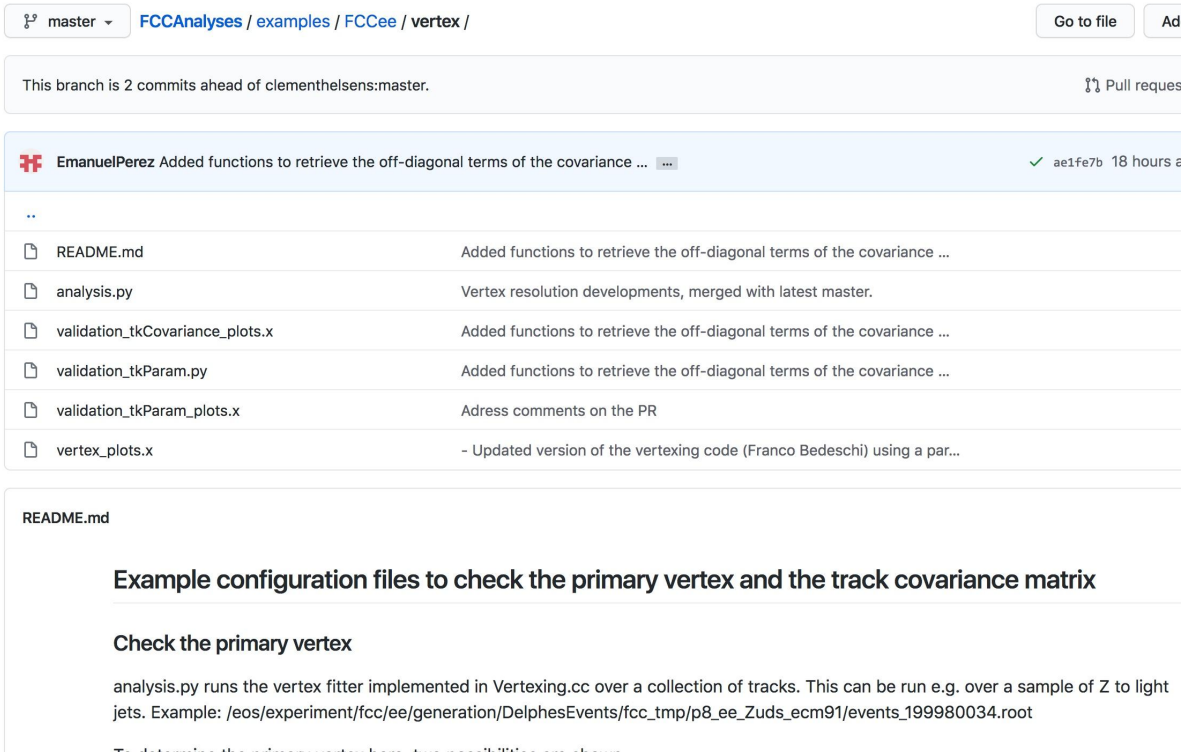






Document the examples - In FCCAnalyses

https://github.com/HEP-FCC/FCCAnalyses/blob/master/examples/FCCee/vertex/



master ▾ FCCAnalyses / examples / FCCee / vertex / Go to file Ad

This branch is 2 commits ahead of clementhelsens:master. Pull reques

EmanuelPerez Added functions to retrieve the off-diagonal terms of the covariance ... ae1fe7b 18 hours a

README.md	Added functions to retrieve the off-diagonal terms of the covariance ...
analysis.py	Vertex resolution developments, merged with latest master.
validation_tkCovariance_plots.x	Added functions to retrieve the off-diagonal terms of the covariance ...
validation_tkParam.py	Added functions to retrieve the off-diagonal terms of the covariance ...
validation_tkParam_plots.x	Adress comments on the PR
vertex_plots.x	- Updated version of the vertexing code (Franco Bedeschi) using a par...

README.md

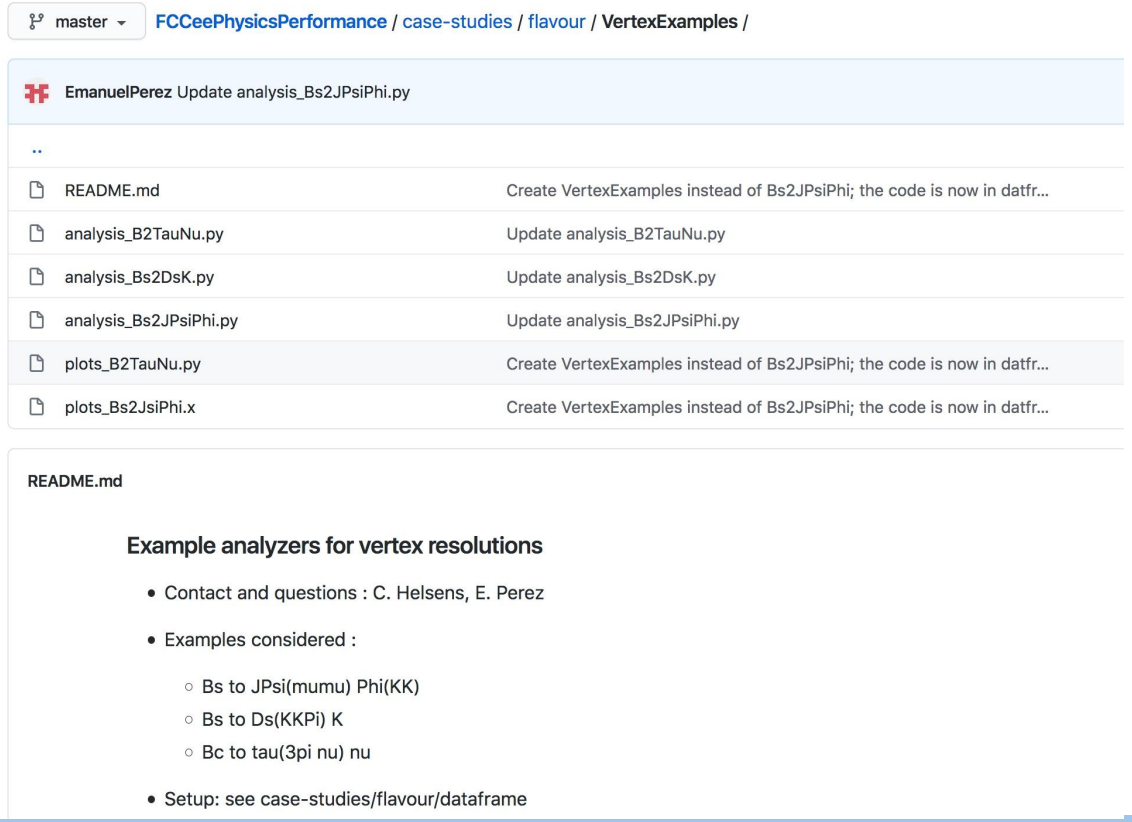
Example configuration files to check the primary vertex and the track covariance matrix

Check the primary vertex

analysis.py runs the vertex fitter implemented in Vertexing.cc over a collection of tracks. This can be run e.g. over a sample of Z to light jets. Example: /eos/experiment/fcc/ee/generation/DelphesEvents/fcc_tmp/p8_ee_Zuds_ecm91/events_199980034.root

Document the examples - In FCCeePhysicsPerf

https://github.com/HEP-FCC/FCCeePhysicsPerformance/tree/master/case-studies/flavour/VertexExamples



master FCCeePhysicsPerformance / case-studies / flavour / **VertexExamples** /

EmanuelPerez Update analysis_Bs2JPsiPhi.py

..

README.md	Create VertexExamples instead of Bs2JPsiPhi; the code is now in datfr...
analysis_B2TauNu.py	Update analysis_B2TauNu.py
analysis_Bs2DsK.py	Update analysis_Bs2DsK.py
analysis_Bs2JPsiPhi.py	Update analysis_Bs2JPsiPhi.py
plots_B2TauNu.py	Create VertexExamples instead of Bs2JPsiPhi; the code is now in datfr...
plots_Bs2JsiPhi.x	Create VertexExamples instead of Bs2JPsiPhi; the code is now in datfr...

README.md

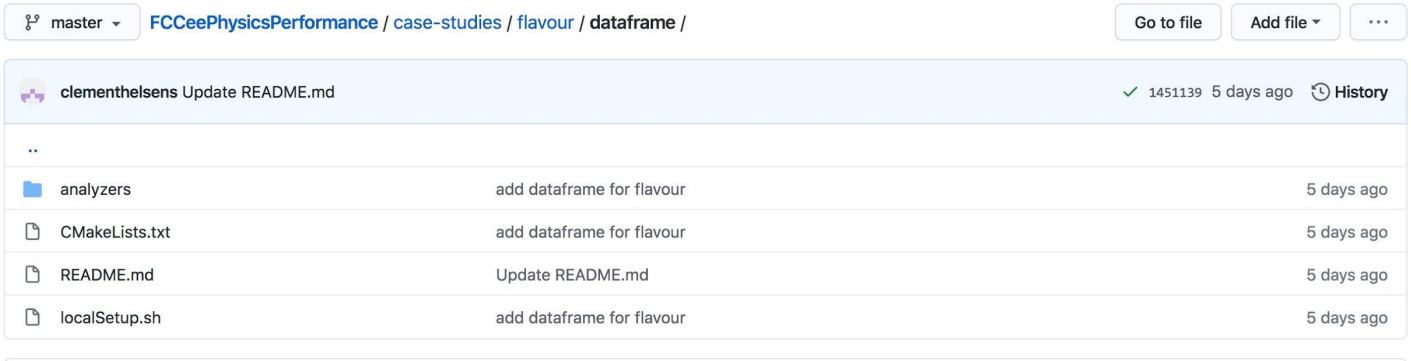
Example analyzers for vertex resolutions

- Contact and questions : C. Helsens, E. Perez
- Examples considered :
  - Bs to JPsi(mumu) Phi(KK)
  - Bs to Ds(KKPi) K
  - Bc to tau(3pi nu) nu
- Setup: see case-studies/flavour/dataframe

Structure the code

- In FCCAnalyses common code structure
  - Lots of developments ongoing
  - Lots of functions available

    -> using namespacing for simplicity and clarity when calling a function, e.g. :
    - Vertexing, VertexingACTS, VertexingUtils
- In FCCeePhysicsPerformance custom code
  - Working example to use custom code from FCCeePP in FCCAnalyses



3. Next steps

Next steps: using the Acts tracking suite

- Vertex finding in Acts factorizes from the Acts tracking
  - Implementation similar to what was done with Franco's code has already been achieved in FCCAnalyses
  - Primary vertexing finding available in FCCAnalyses
  - Vertex resolutions studies can be compared with the results obtained from Franco's code
  - Comparisons can be made between the different algos that are in ACTS; performances in view of b-tagging can be studied (multi-vertex finding)
  - Integrate Acts primary vertex finder in key4Hep so that PV reconstruction is done upstream
  - Acts has a lot of other useful tools that we could use
    - Vertex fitting, track extrapolation
    - could very easily use Acts vertex fitter and compare results with FB code

Next steps: using the LCFI+

- LCFI+ (the algorithm used by LC)
  - Can be run over EDM4Hep files via a “wrapper” of the source code, and a transient, on-the-fly, conversion of the EDM4Hep event into the LCIO format.
  - Good progress recently on this conversion !
- Finalize the SW implementation:
  - Need to convert back the vertices found by LCFI+ , from LCIO to EDM4Hep
  - Could be quick. Maybe we can have the LCFI+ vertices in the Monte-Carlo samples that we’ll produce within the coming month
- Comparisons of LCFI+ vertices with vertices from FB’s algo or Acts
- Could use the LFCI+ displaced vertices in b- or c-tagging algorithms
- Bonus: LCFI+ does not only find vertices, that’s actually a b- / c- tagging algo.
  - Train the BDT for FCC-ee
  - Compare performances with the ParticleNet algorithm from Loukas & Michele

Next steps: using Franco's vertex fitter

- Stand-alone code from Franco Bedeschi
  - Only dependencies = ROOT and Implemented in FCCAnalyses
  - Already used for first estimations of the expected resolutions on DV in chosen examples
  - Follow the updates (beam spot constraints, covariance matrix propagation, etc...)

- Consolidate and pursue these studies
  - Estimation of the effect of variations of the detector model: change the radii of the layers of the vertex detector, the thickness of the layers or the BP, the single-hit resolution: can be done with the tools that are in place !
  - Volunteers have already contacted us, instructions are ready and have been sent

- Try the same setup to study reconstruction of far-detached vertices
  - Write a vertex-finding algorithm that runs the fitter iteratively, to determine first the primary vertex, and then displaced vertices, in order to tag "b-jets" inclusively.
    - Could be started now too.

Next steps: using the DecayTreeFitter

- Global fit of a whole decay chain(Babar, LHCb, Belle-2)
  - Implementation has not started yet.
- Implementation : may actually not be too difficult / long.
  - Start from LHCb implementation
  - Wouters: “a few months should be more than sufficient to (re)implement it”
  - Very nice opportunity for someone to start a standalone DTF implementation in Key4Hep based on:
    - LHCb implementation, possibly some Babar code, but could in principle use Acts core components

- Once done
  - Opens the door to many dedicated studies in the area of flavour physics
  - by how much DTF improves the sensitivity, e.g. in B to K* τ τ for which a very precise determination of the vertices is crucial.

4. Summary

Summary - 1

- Detailed summary of the meeting presented today
  - The meeting was well attended, and the feedback received was very positive
  - Some new contacts established
  - Benefit from the recent progress to kick off a lot of studies that needs vertexing

- Already nice progress made the last month
  - Well documented and validated examples
  - Countless opportunities for new

Summary - 2

- Things are moving fast but
  - Still with very very limited (but efficient) contributors

- Follow-ups
  - Expect follow-ups in future SW and Physics Performance meetings
  - Meanwhile, an informal informal e-group has been created. It we be will use to
    - Communicate additional information
    - Circulate follow-ups on the points discussed during the meeting of last week
    - Possibly call working meetings or discussions
    - vertexing-FCCee-informal
    - Has been announced to the two mailing lists used to announce the topical meeting

Bonus

FCCAnalyses - 1

https://github.com/HEP-FCC/FCCAnalyses/

Common tool for analyzing large datasets using RDataFrame and produce flat ntuples

It is composed of a library of C++ analysers and python configurations files

- C++ analysers are developed in common
- Python code specific to the analysis to define the analysers, output variables, input samples

Flat ntuples are then used for example to:

- Produce variables for MVA training
- Produce final variables for analysis and plotting
- Run decay selector for flavour physics
- Etc...

FCCAnalyses - 2

https://github.com/HEP-FCC/FCCAnalyses/

- Set of tools to help processing the output of ‘simulation’
  - Agnostic to the type of simulation but specific reader functions are required
  - Build a common set of utility functions, algorithms for common use
  - Still possible for users to test their algorithms locally before publishing them
- FCCAnalyses structure