

# GenFit implement into the J-PARC E42 K1.8 Analyzer

2022.07.15

## Environment setting

Nothing to do. Same as the K1.8-analyzer

## Features

- GenFit package
- GenKEK codes
- GenKEK dsts
- HypTPC geometry GDML

## Install

Install the GenFit package and the GenKEK follows.

How-to Compile:

```
$> cp Makefile.genfit Makefile
$> make
$> make pcms
```

You can compile separately 1. K1.8-Analyer(k18ana) or 2. GenKEK(genfit)

```
all: k18ana genfit
k18ana: lib usr dst
genfit: genkek genfit_dst
```

## Library

lib/libK18Analyzer.a : for K1.8-analyzer compiling

lib/libGenKEK.a : for K1.8-analyzer & GenFit package & GenKEK compiling

## Paramters

Add “TPCGDML” in the conf param file

```
TPCGDML: param/geometry/hypypcGeo.gdml
```

And add “Fitter” and “Iteration” in the USER param file

```
#GenFit
Fitter 0 #KalmanFitterRefTrack
#Fitter 1 #KalmanFitter
#Fitter 2 #DAF w/ RefTrack
#Fitter 3 #DAF w/o RefTrack
nIteration 5 20
```

## dst rule

GenKEK dsts have name starting with “Genfit” and should be placed at “dst/”  
e.g.) dst/GenfitHelixGeant4.cc, dst/GenfitSkeleton.cc

# GenKEK development guide

## GenKEK directory

All files are placed in “genfit/genkek/”.  
e.g.) genfit/genkek/include/HypTPCTask.hh, genfit/genkek/src/HypTPCTask.cc

## Units

GenFit : GeV/c, ns, cm, kGauss  
K1.8Ana : GeV/c, ns, mm, T

## Features

Environment setting and translation

- HypTPC geometry file (param/geometry/hytpcGeo.gdml)
- HypTPCFieldMan : HS field management
- HypTPCSpacePointMeasurement & HypTPCHit : Translating Hit-Pos&Resolution information into the GenFit format
- HypTPCFitter : Handling GenFit Fitting algorithms

Main Parts for the track fitting

Inheritance ( HypTPCTrack -> HypTPCFitProcess -> HypTPCTask )

- HypTPCTrack : Track container  
GenFit tracks can be provided several TrackReps to describe the same track in order to fit different particle hypotheses(Pion, Proton ...) -> Find best result (default setting is finding minChi2)

TPCLocalTrack/TPCLocalTrackHelix should provide “PDGcode” and initial “Position seed” & “Momentum seed” from pre-fitting.

- HypTPCFitProcess : Handling the fitting process
- HypTPCTask : Useful functions (Handling the fit results)  
e.g.) get fitting results (chi2, ndf, tof, mom, length, residuals...) or useful functions (extrapolation...)

## Development guide

Please add more functions in the HypTPCTask class.

You can get a fitted-track from the container or TrackRep or FitState or FitStatus.

```
genfit::Track* GetFittedTrack(int trackid) const;
genfit::AbsTrackRep* GetTrackRep(int trackid) const;
genfit::FitStatus* GetFitStatus(int trackid) const;
genfit::MeasuredStateOnPlane GetFitState(int trackid) const;
```

Please use the GenFit functions to make it work the way you want.

You can find most useful GenFit funtions in the follows.

Expecially, core directory has most useful functions.

All header files have discription.

```
genfit/core/include/track.h .. : useful track functions
genfit/core/include/MeasuredStateOnPlane.h, StateOnPlane.h ... : functions for State vector
genfit/core/include/AbsTrackRep.h : extrapolation and others
genfit/core/include/FitStatus.h : functions to call the fitstatus
```

And there are functions to print the fitting parameters or extrapolation and the others

```
//Parameters
double GetChi2(int trackid) const;
double GetNDF(int trackid) const;
double GetChi2NDF(int trackid) const;
double GetCharge(int trackid) const;
TVector3 GetMom(int trackid) const;
TVector3 GetPos0(int trackid) const; //Get Vertex position
double GetTrackLength(int trackid, int start=0, int end=-1) const;
double GetTrackTOF(int trackid, int start=0, int end=-1) const;
```

```
//Extrapolation
bool ExtrapolateTrack(int trackid, double distance, TVector3 &pos) const;
bool ExtrapolateToPoint(int trackid, TVector3 point, TVector3 &pos) const;
bool GetPosOnPlane(int trackid, genfit::SharedPlanePtr plane, TVector3 &pos) const;
bool IsInsideTarget(int trackid) const;
```