Tracker Software Status

P .Zuccon – INFN Perugia

AMS Analysis/Software Meeting

October 11th, KSC

INFN
Istituto Nazionale
di Fisica Nucleare

Overview

- Many improvement since the end of the test beam
  - Alignment second pass almost available
  - Easier access to the track info
  - Offline (re)fit of the tracks
  - Improved efficiency of matching with other subdedectors
  - Many (!) bugfixes
  - review and speed up of the facilities to handle the events (AMSChain & C.)
  - Implementation of the tracker part for the "MC raw data
  - Tuning of the Si signal simulation (A. Oliva)

P.S.: Thanks to Vitaly, for pointing out problems, suggesting solutions and pushing for fixes …

Easier Access to the Fit info

- The most relevant information for the track is the fit result
- The info about a TrTrackR fit are kept in a TrTrackPar Object
- Track Fitting can be done in multiple ways depending on: the fitting algorithm or the subset of hits involved
- To uniquely identify a fit we use the 32 bits of an integer(fit_code) as flag for the attributes
- A new method has been developed to generate this fit_code using a human readable syntax
- Currently there are:
  - 3 algorithms: kChoutko, kAlcaraz, kChikanian
  - 5 types: all hits, Internal, upper Half, lower Half, External only
  - Multiple scattering switch
- The new implemented TrTrack method is called iTrTrack(…)

iTrTrack method

```
int TrTrackR::iTrTrackPar ( int  algo = 0,
                            int  pattern = 0,
                            int  refit = 0
                          )
```

It gives you the integer number (fit code) to be used to access the fit results (**TrTrackPar** obj) Advanced **TrTrackPar** accessor.

Parameters:

algo Fitting algorithm= 0 The default algorithm choosen at recon stage 1 Choutko; 2 Alcaraz; 3 Chikanian; +10 mscattering off;
pattern Hit Pattern= 0 all hits; 1 upper half; 2 lower half; 3 drop 2 external hits; 4 only 2 + 2 external hits;

OR

mmmmmmmmm where m=0 or 1 for TrRecHit layer GetLayer() from right to the left such as 100110010 corresponds to layers 2,5,6,9

Parameters:

refit 0 do not refit 1 refit if does not exist 2 refit

Returns:

the code to access the **TrTrackPar** object corresponding to the selected fit or -1 if not found

Example of TrTrack Methods that use fit_code

	Returns Chi2 on X.
double	**GetChisqY** (int id=0) *Returns Chi2 on Y.*
double	**GetChisq** (int id=0) *Returns Global Chi2.*
int	**GetNdofX** (int id=0) *returns Ndof on X*
int	**GetNdofY** (int id=0) *returns Ndof on Y*
double	**GetRigidity** (int id=0) *Returnt the fitted Rigidity.*
double	**GetErrRinv** (int id=0) *Returns the error on 1/R.*
AMSPoint	**GetPentry** (int id=0) *Get track entry point (first layer of the fitting).*
AMSPoint	**GetP0** (int id=0) *Returns the point of passage on the Z=0 XY plane.*
AMSDir	**GetPdir** (int id=0) *Get track entry point direction (first layer of the fitting).*
AMSDir	**GetDir** (int id=0) *Returns the direction at the point of passage on the Z=0 XY plane.*
AMSPoint	**GetResidual** (int ilay, int id=0) *Return the 3D residual at layer ilay (0-7).*
char *	**GetFitNameFromID** (int fitnum) *Get back the string corresponding to a fit ID.*
int	**GetFitID** (int pos) *Get the fit ID of the pos-th fit method or zero if pos is invalid.*
void	**PrintFitNames** () *Print the string IDs of all the performed fits.*
AMSPoint	**GetPlayer** (int ilay, int id=0) *Get track position at layer ilay (0-7).*

TrTrackPar Object

Public Attributes

bool	**FitDone** *Fit done flag.*
short int	**HitBits** *Bits of hits used for the fitting e.g. 0x7f: Layer1 missing.*
Double32_t	**ChisqX** *Chisquare in X (Not normalized).*
Double32_t	**ChisqY** *Chisquare in Y (Not normalized).*
short int	**NdofX** *Ndof in X.*
short int	**NdofY** *Ndof in Y.*
Double32_t	**Chisq** *Normalized chisquare, Chisq(x+y)/Ndof(x+y).*
Double32_t	**Rigidity** *Rigidity in GV.*
Double_t	**ErrRinv** *Fitting error on 1/rigidity in 1/GV.*
AMSPoint	**P0** *Track positon (normally defined at Z=0 plane).*
AMSDir	**Dir** *Track direction at P0.*
AMSPoint	**Residual** [trconst ::**maxlay**] *Fitting residual at each layer.*

Offline refit

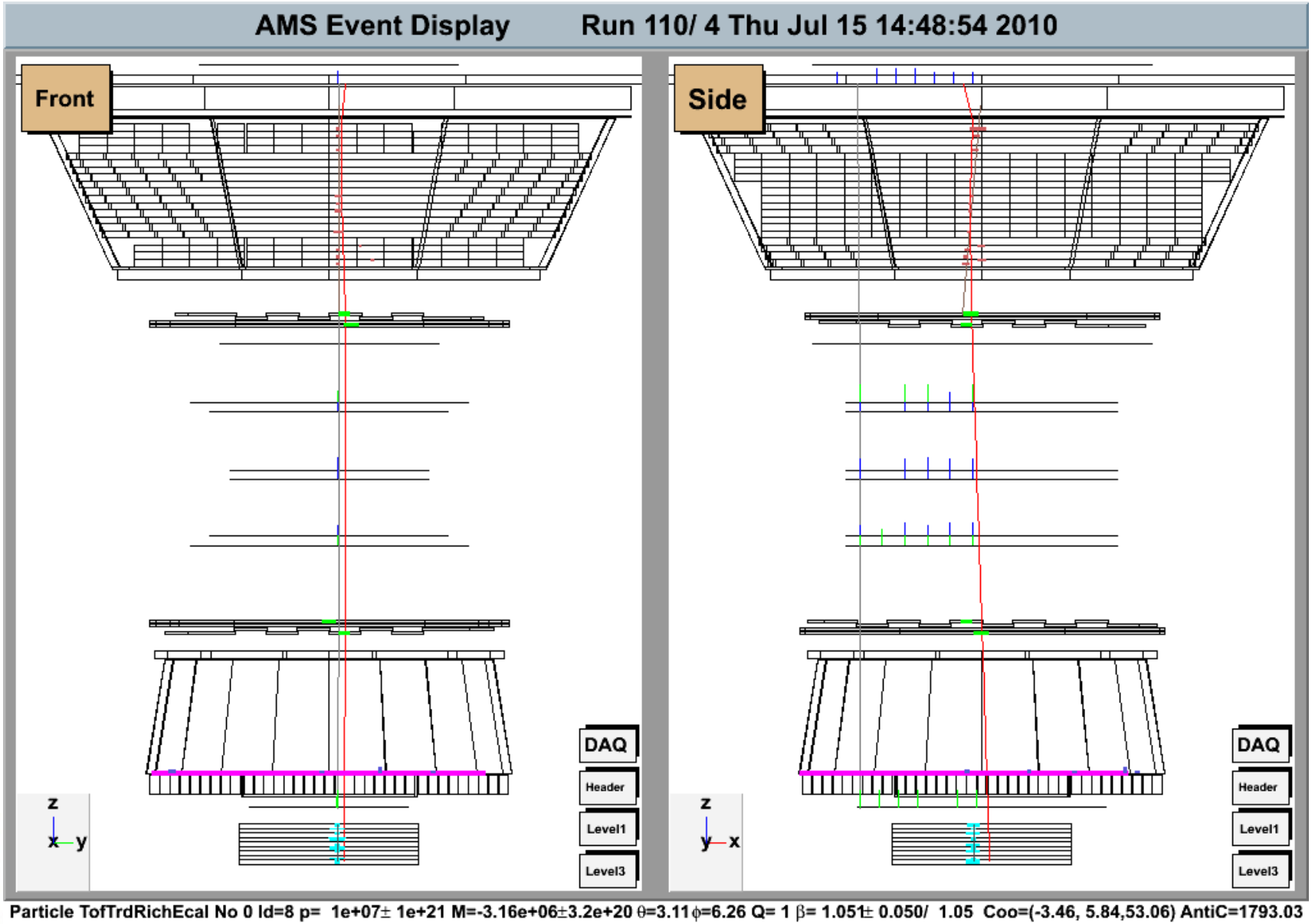
- Via the iTrTrack method it is possible to refit the track from the root files.
- The Magnetic Field map file is needed and the program automatically search for the file `$AMSDataDir/v5.00/MagneticFieldMapPM_NEW.bin`
- (temporarily) An initialization is need at the begin of the program:

Refit Code example

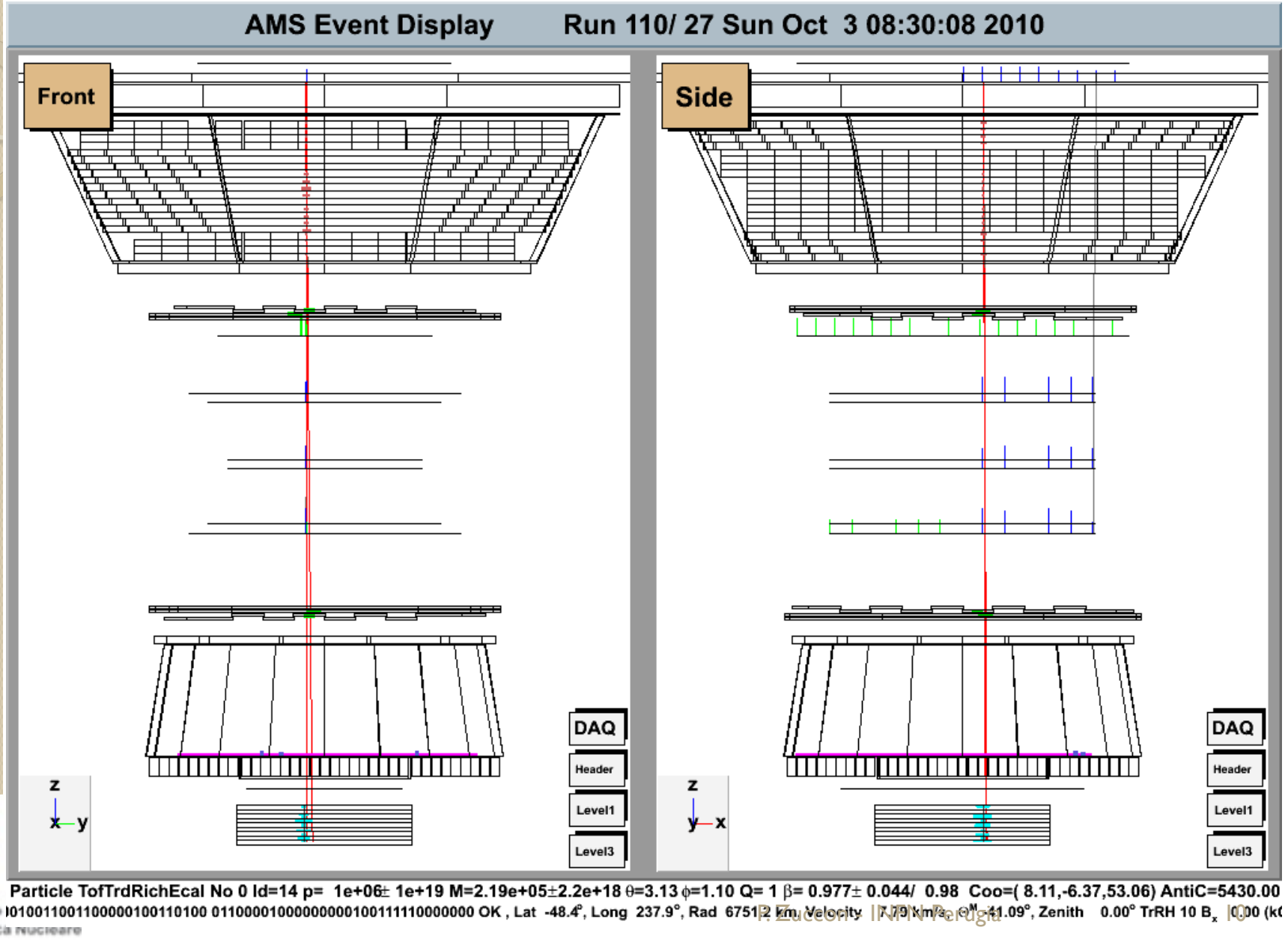
```
#include "root_RVSP.h"
#include "amschain.h"
#include "MagField.h"
#include "tkdcards.h"


int main(){
  AMSChain* ams = new AMSChain();
  ams->Add("/r0fc00/Data/AMS02/2009B/
            bt.aug2010.v5/pr400/1281546656.00000001.root");
  TRFITFFKEY.init();
  AMSEventR* ev= ams->GetEvent();
  TrTrackR* nn=0;
  while(1){
    ev= ams->GetEvent();
    if(ev->nTrTrack()>0 ){
      nn=ev->pTrTrack(0);
      if(nn->getnhits()==9) break;
    }
  }
  printf("\nChoutko MS Refit\n");
  typ=nn->iTrTrackPar(1,0,2);
  pp=nn->gTrTrackPar(typ);
  pp.Print(1);
  return 1;
}
```

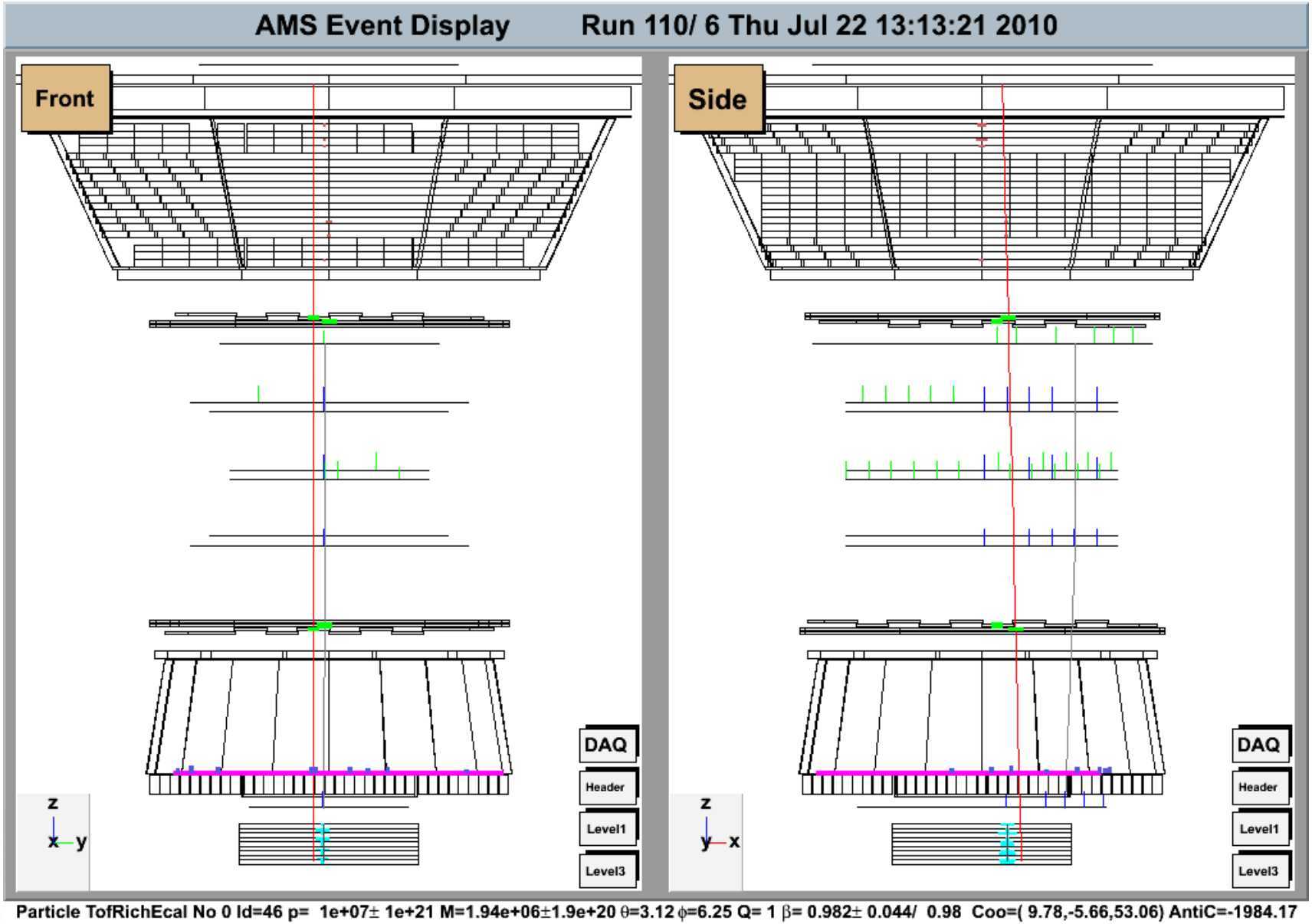
Tracking efficiency



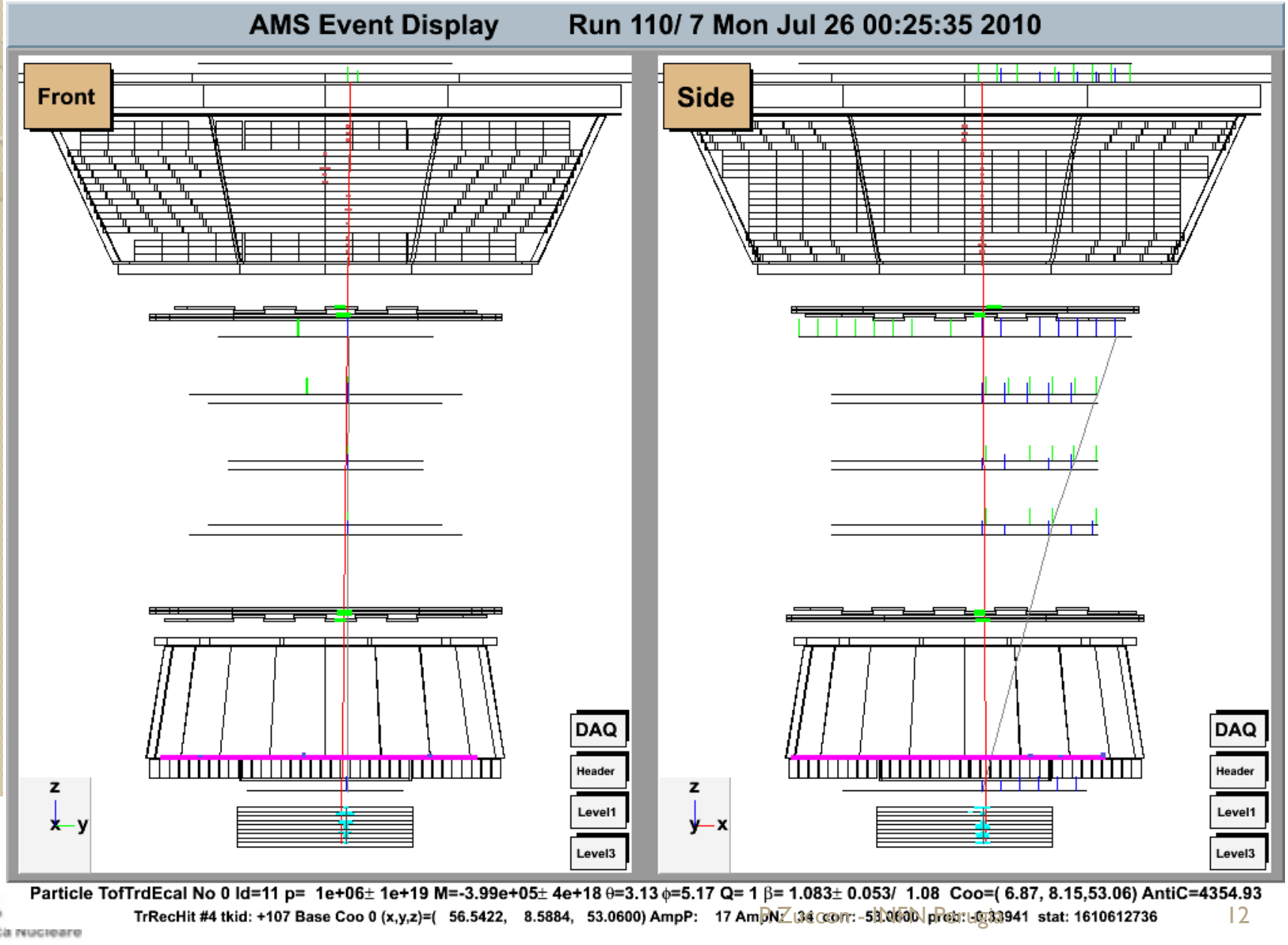
Tracking efficiency



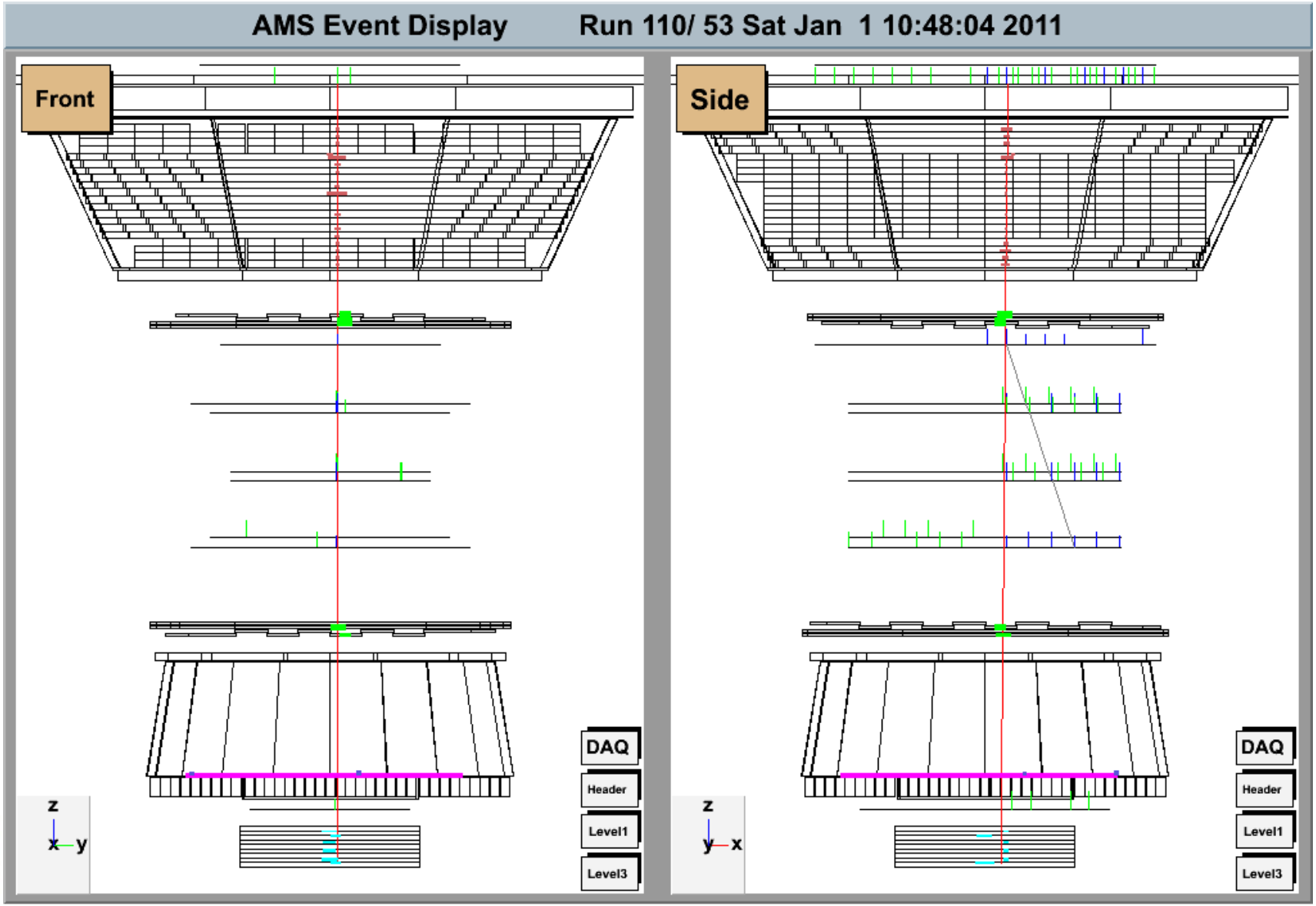
Tracking efficiency



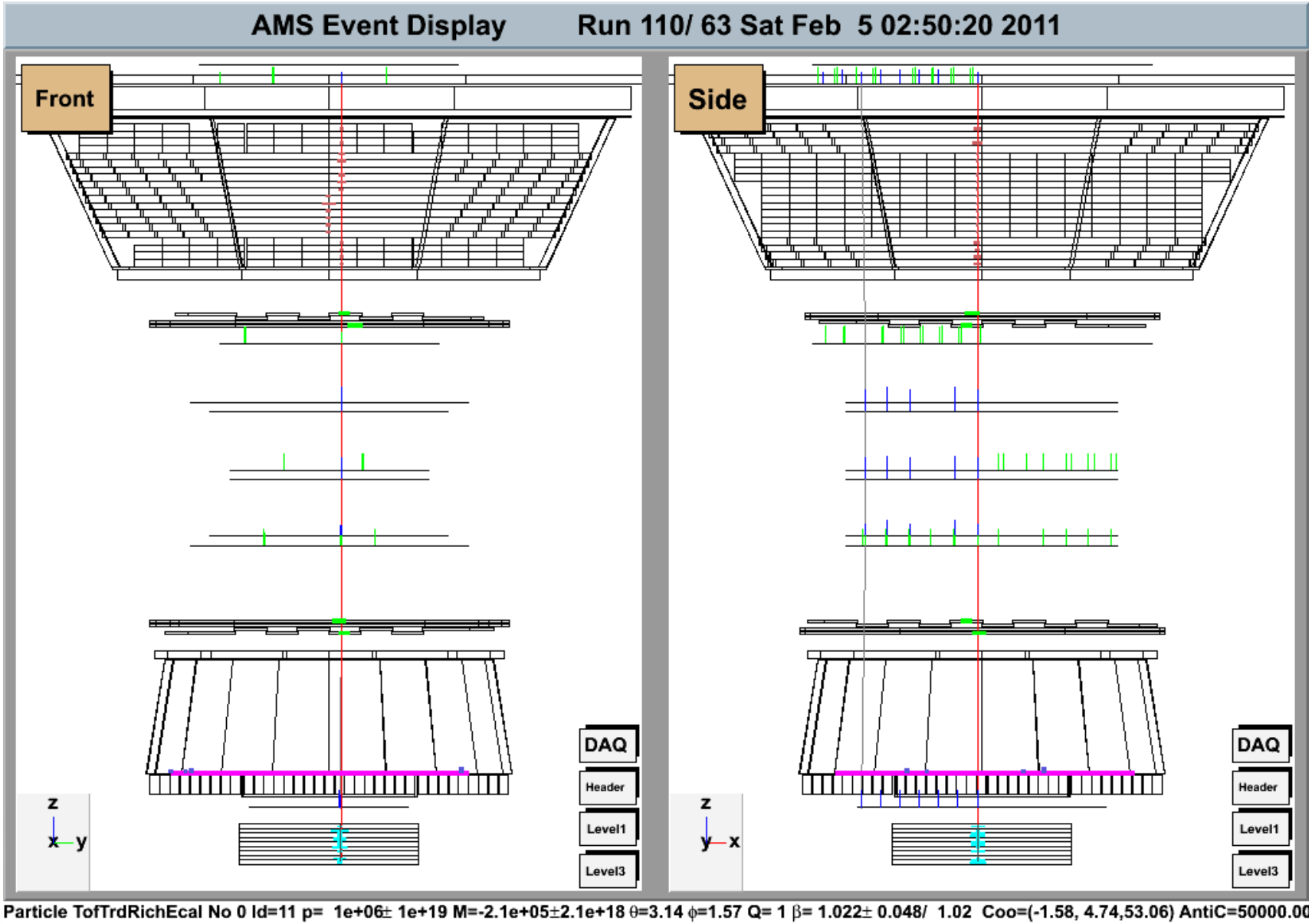
Tracking efficiency



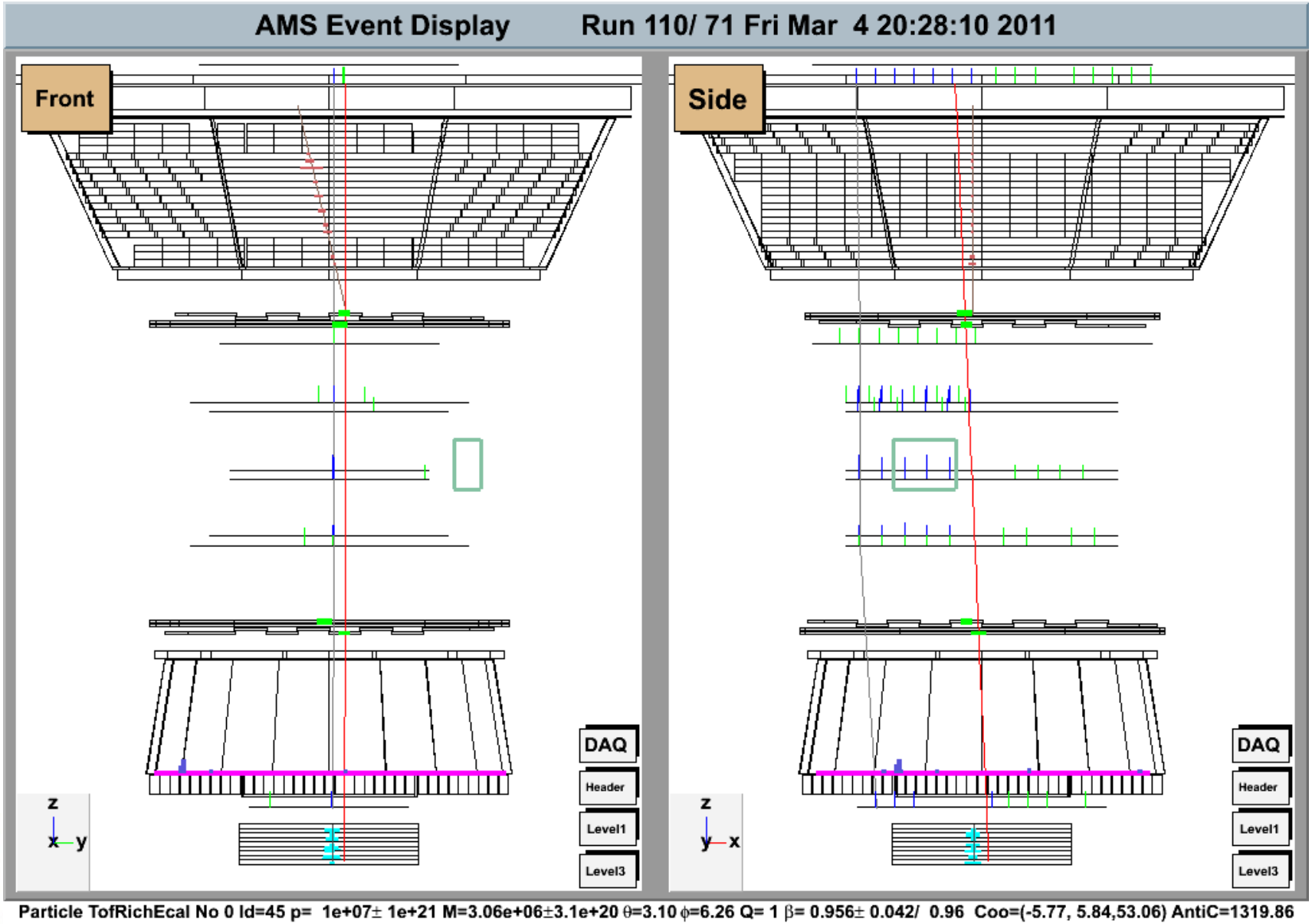
Tracking efficiency



Tracking efficiency

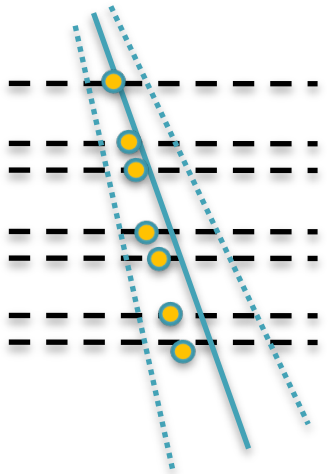


Tracking efficiency



Solution

A. Use TRD track
   - if a trd track has a good match on Y coo and on direction
   - extrapolate the TRD track to the tracker region
   - for each XY TrRecHit check if a multiplicity is close to the trd track within ERR_TRD
   - if all the XY hit of the track satisfy the previous requirement fix the X coo of the Y only hits and refit the track



B. Use a TOF "track"
   - if the TrTrack has a good Y match on TOF planes 1 and 4 repeat the same procedure of TRD but with ERR_TOF

C. The procedure is further improved by increasing ERR_xxx as a function of the Tracker Layer number

Test of Tracking efficiency

Generate events that are likely to be reconstructed:

- No Hadronic interaction in GEANT3
- 10 GeV protons
- fire form a 10 x 10 cm area pointing downward

	No_Tk	Tk & NoTkP	Tk & TkP
No Match	25%	38%	36%
TRD only	25%	13%	61%
TRD + Tof	25%	2%	73%

Improvements to AMSChain

- Some are just hidden to the normal user but they affect the performance
- Tested with TProofLite soon new release of AMSRoot
- Save selected events to a different file

void	**OpenOutputFile** (const char *filename) *Opens the file to output selected events.*
void	**SaveCurrentEvent** () *Saves the current entry to the output file (if it is not open, it creates SelectedEvents.root).*
void	**CloseOutputFile** () *Properly closes the Output File for selected events.*

Summary

- Many improvement since the end of the test beam
  - Alignment second pass almost available
  - Easier access to the track info
  - Offline (re)fit of the tracks
  - Improved efficiency of matching with other subdedectors
  - Many (!) bugfixes
  - review and speed up of the facilities to handle the events (AMSChain & C.)
  - Implementation of the tracker part for the "MC raw data
  - Tuning of the Si signal simulation (A. Oliva)
- To be done soon
  - Optimize the track fitting considering the materials
  - Implement the reconstruction for track pointing to ACC
  - Review code that implements the matching among detectors to avoid duplications
  - Implement the charge reconstruction for the tracker
  - Test, test and test