FTPC Analyzer

LC-TPC R&D Meeting
Berkeley
18.-19. October 2003

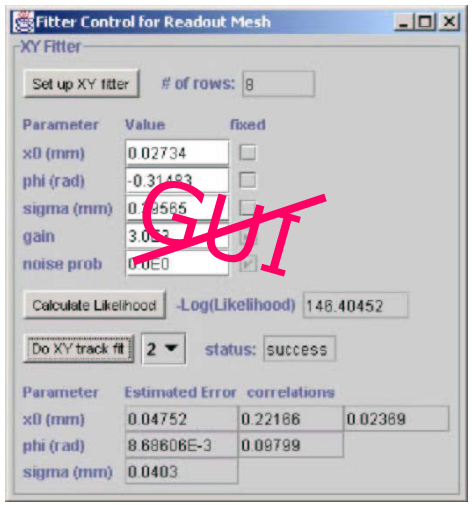
Kirsten Sachs
Carleton University

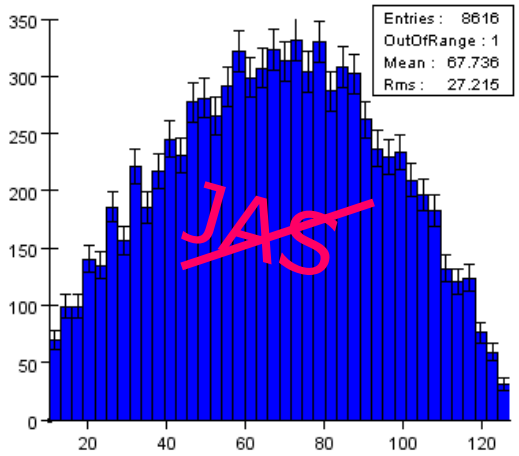
A program package for TPC analysis

- signal reconstruction (amplitude, T0)
- track finding and fitting

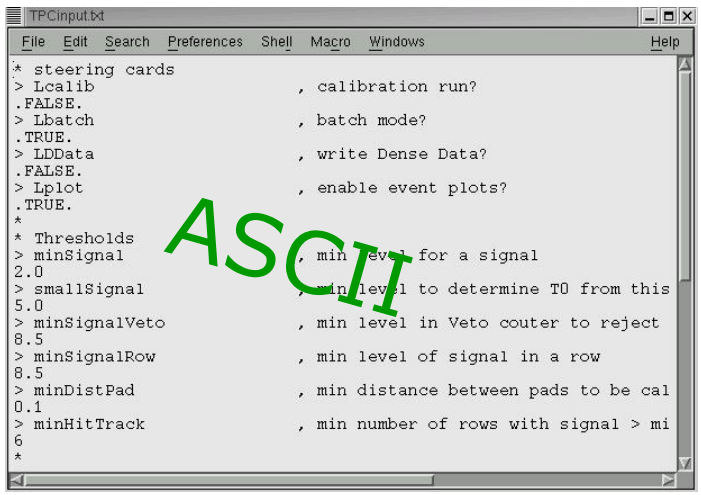
The Language

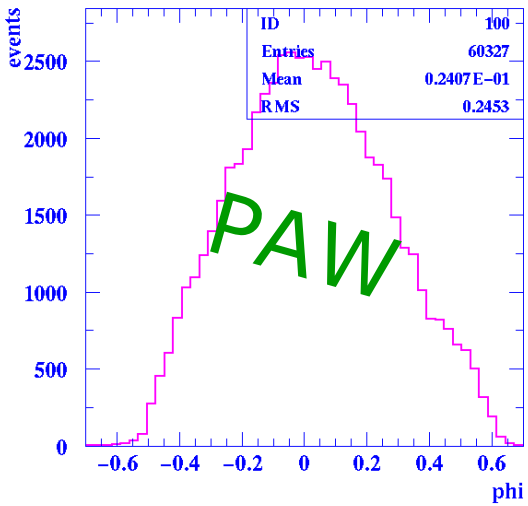
It's not JAVA





It's FORTRAN





F ≅ f95 /= f77

MYanalyzer.f95

File Edit Search Preferences Shell Macro Windows Help

```
MODULE MYanalyzer
   USE IOunits
   USE Threshold
   USE TPCdata
   USE TPCpaw
   USE TPCcalcAmplitude
   USE TPCtrack
   USE Space

   PRIVATE

   PUBLIC  :: analyzEvent,MY_book
   PRIVATE :: NF_analysis,HF_test

CONTAINS

   SUBROUTINE analyzEvent(iEvent)
      INTEGER,INTENT(IN) :: iEvent
      INTEGER :: IVETO,IERR
      REAL :: time0,deltaT,Amplitude,AmplVeto,T,dwith5,dwout5
      TYPE(Line_type) :: seedTrack
      TYPE(Track_type) :: fitTrack
      INTEGER,DIMENSION(8) :: iRowT=(/1,2,4,5,6,7,9,10/)
      ....
      REAL(KIND=spPrec) :: X0Fit,PhiFit,SigmaFit

! get the signal amplitudes
      CALL calcAmplitudeRow()
      CALL HF_amplitude(iRowT)

! correct the gain
      CALL correctGain()

! preselection
      CALL eventVeto(iGroupV,iRowSeed1,iRowSeed2,iRowT,IVETO)
      IF( IVETO /= 0 )RETURN

! find seed track and NF for event display
      CALL findSeedTrack(iRowSeed1,iRowSeed2,iRowT,seedTrack,IERR)
      IF( IERR /= 0 ) RETURN

! X-Y track, NF for event display
      CALL TrackFitWidth(iRowT,seedTrack,fitTrack)
      CALL NF_event(fitTrack)

      CALL getTrack(fitTrack,X0Fit,PhiFit,SigmaFit)
```

User routine
event loop

optimized
not for speed
but for flexibility

pad shapes:
rectangles
grid (arbitrary)
chevron (soon)

Data formats:
MIDAS
JTPC Monte Carlo
DenseData
Saclay STAR (planned)
… …

no curved tracks (yet)

Basics

- initialization
  read input files, layout, allocate arrays, book histograms, …
- calibration run
  determine pedestals, relative gain, fall-times, rise-times
  from average as function of group or drift time
- write out DenseData
  #group, amplitude, T0 for channels with signal
- analysis
  determine signal amplitudes
  find seed track
  Y-Z track fit (Z : drift distance)
  X-Y track fit (X-Y : pad plane)
  resolution fits
  fill ntuples and histograms

multiplexing
allowed also across rows

don't need it?
iPad = iGroup

Requirements

Compiler: f95 or F (free):
http://www.fortran.com/imagine1

CERNLIB:
PAW, MINUIT

Code:
http://www.physics.carleton.ca/
~sachs/TPC/F

Interface to your data:
F (f77) or C

At least 3 input files:
TPCinput.txt
TPClayout.txt
TPCfiles.txt

TPCinput.txt

```
* steering cards
> Lcalib            , calibration run?
.FALSE.
*
* Thresholds
```

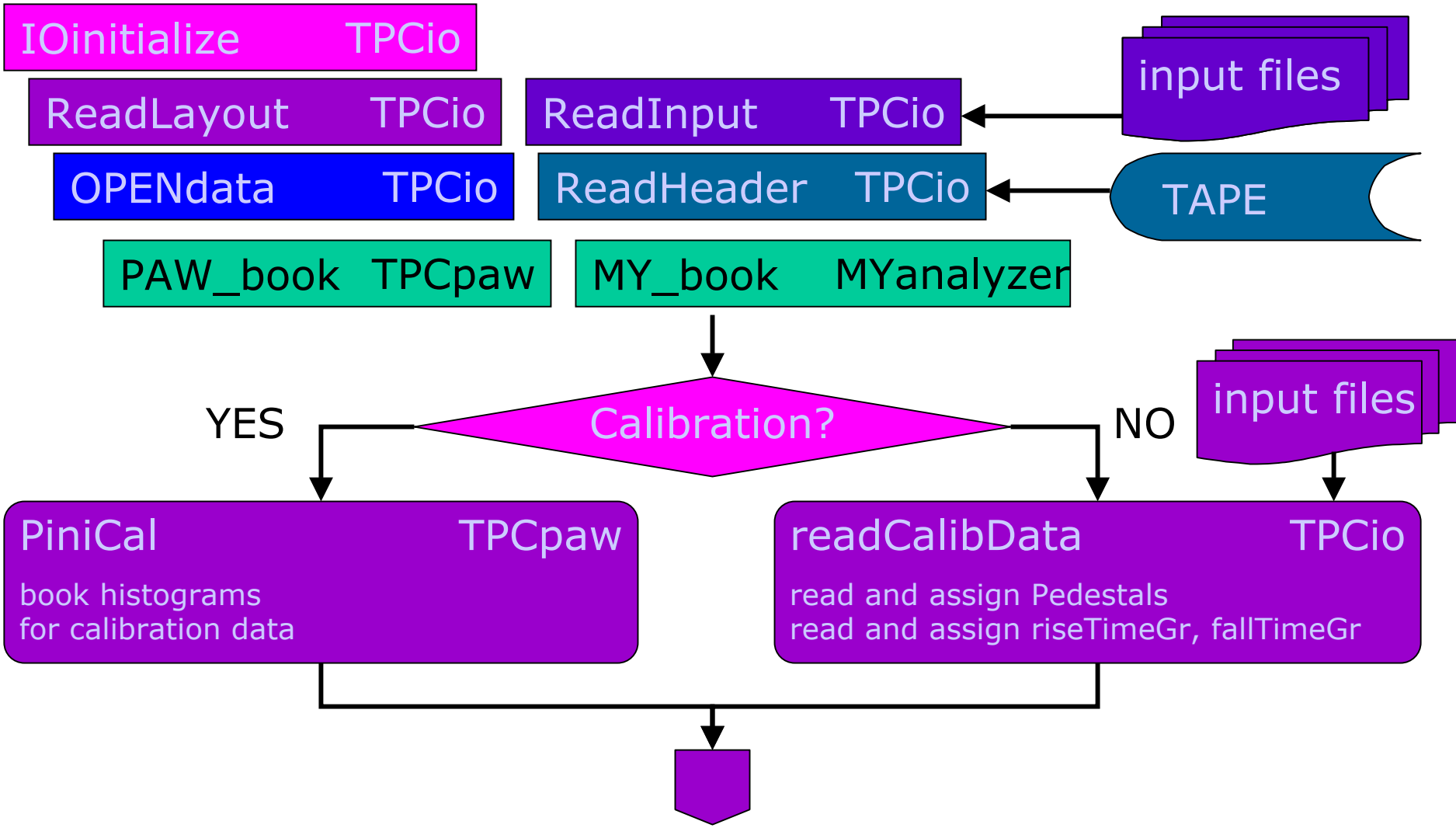
TPClayout.txt

```
192    Pads
 64    Groups    (readout channel)
```

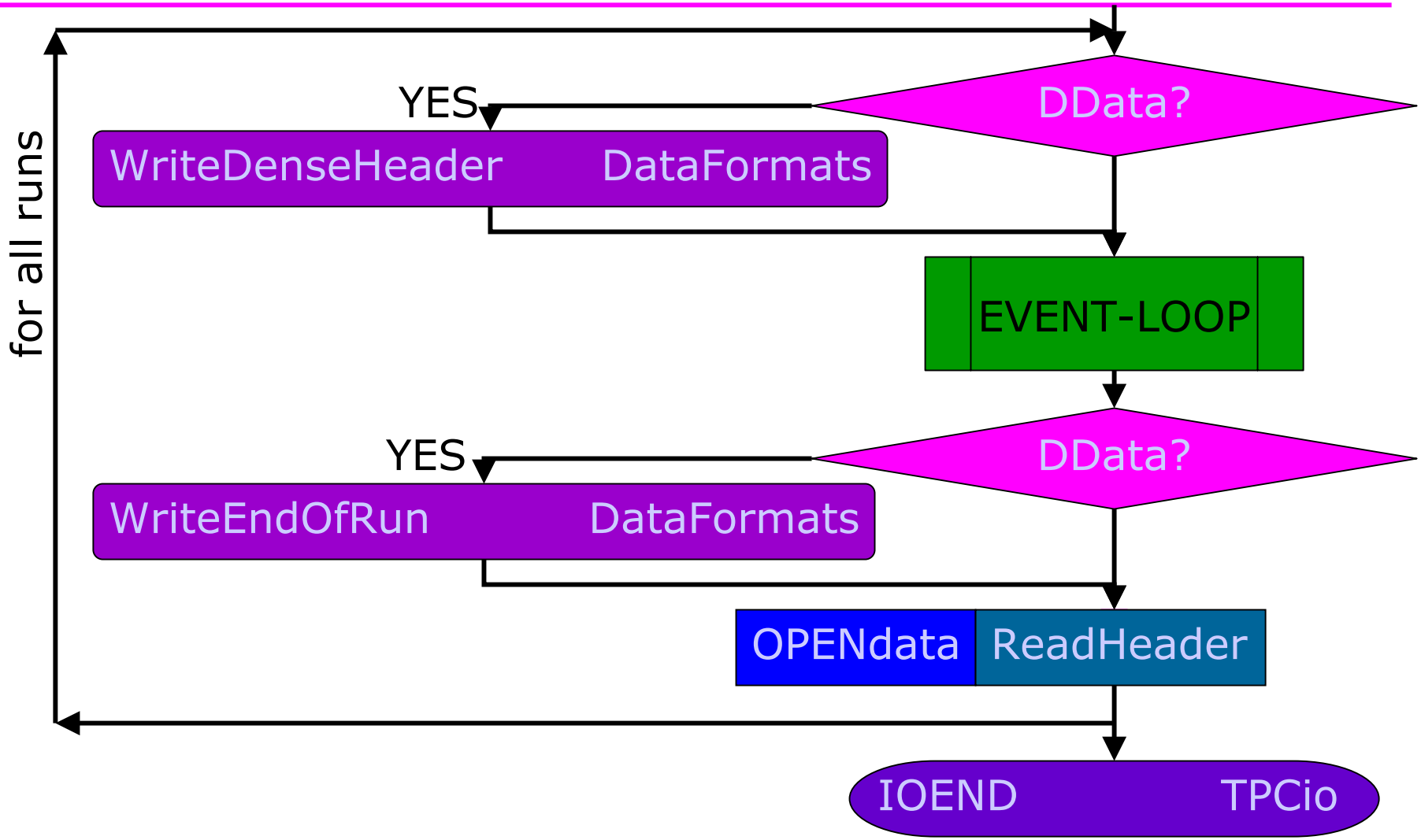
TPCfiles.txt

```
F 1              ! number of runs
* R 1 200          ! range of events
S 217 411 743    ! sample of events
> MIDAS files:
/files2/data/tpc/run01.mid
/files2/data/tpc/run02.mid
```

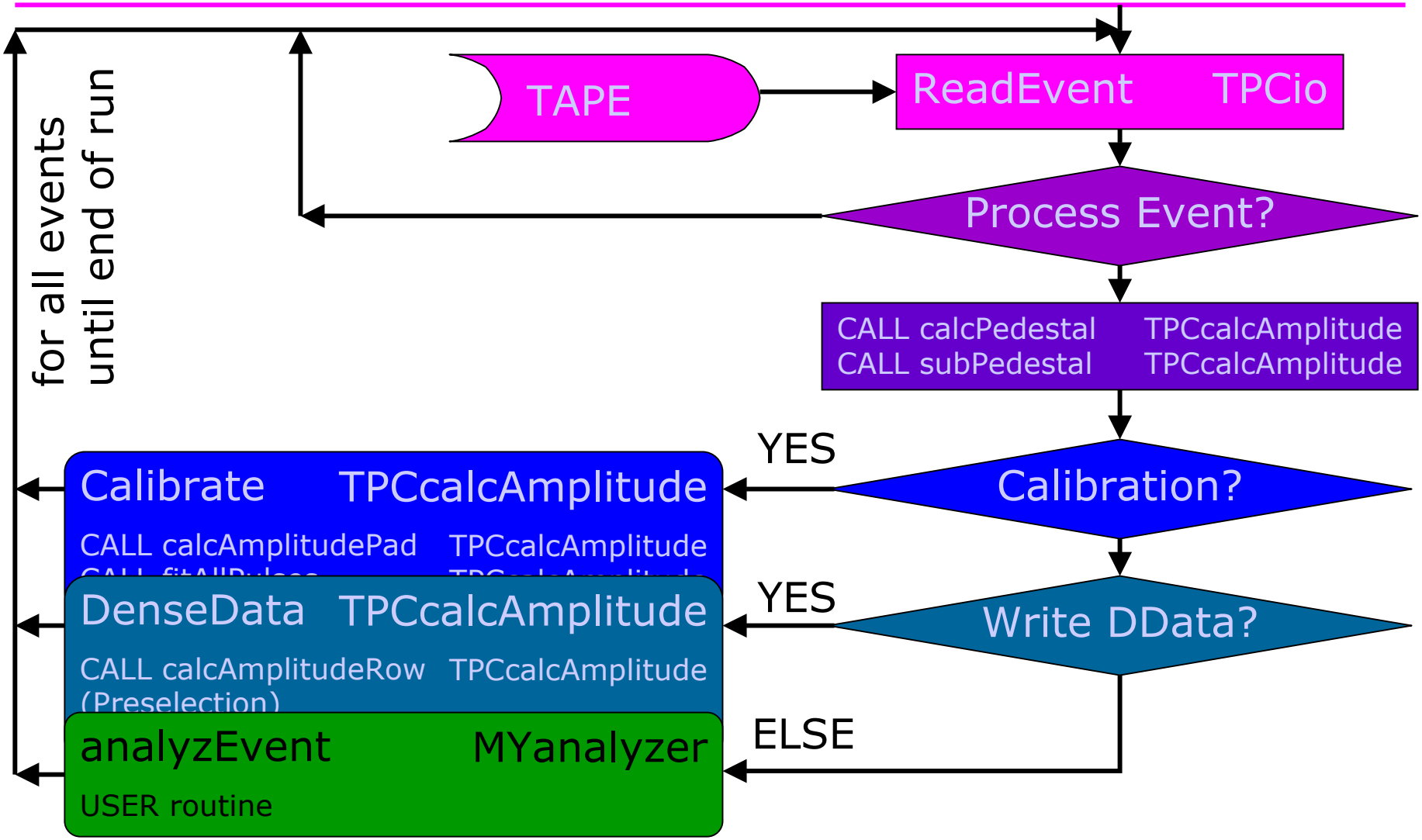
Program-Flow I



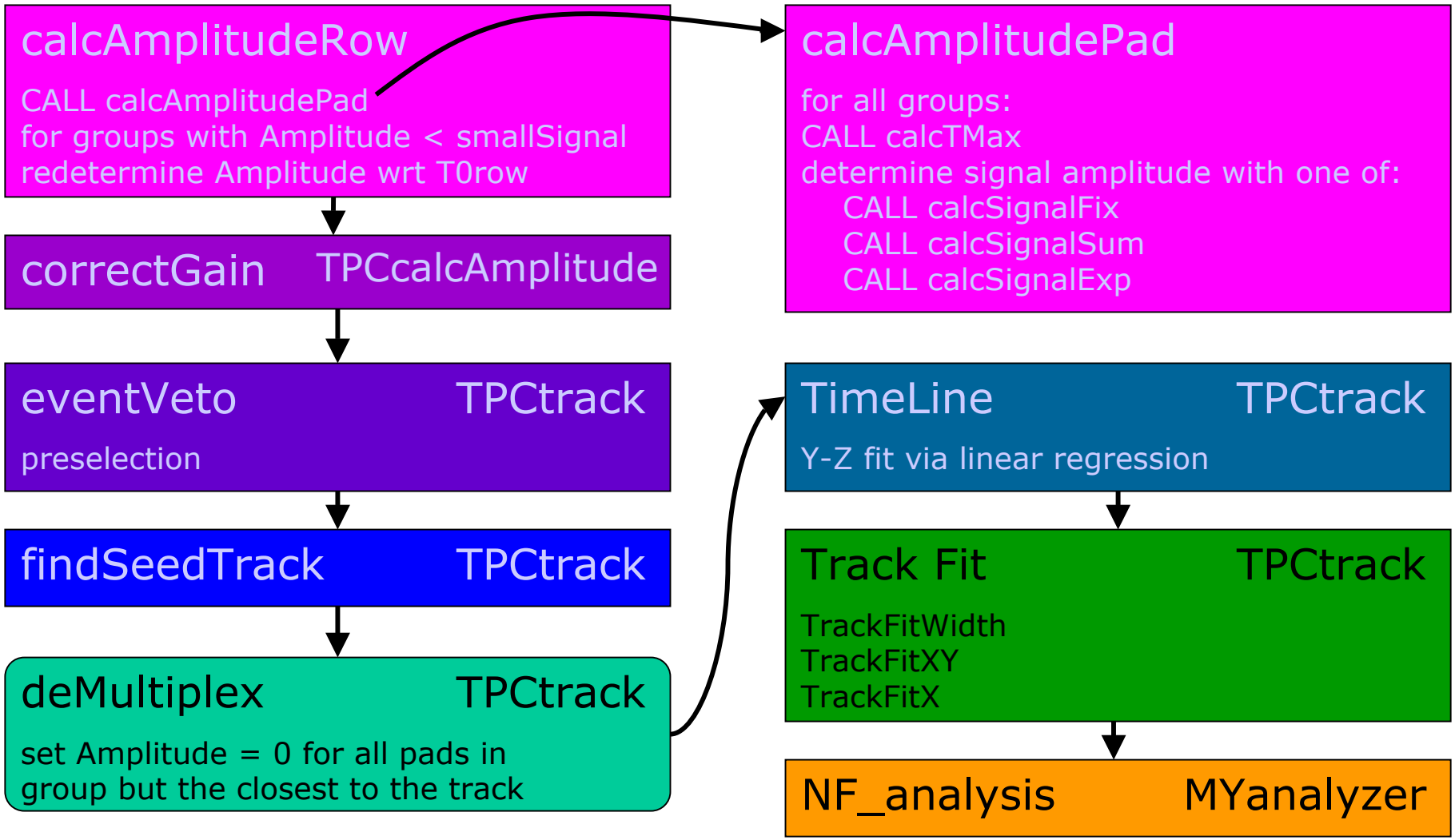
Program-Flow II

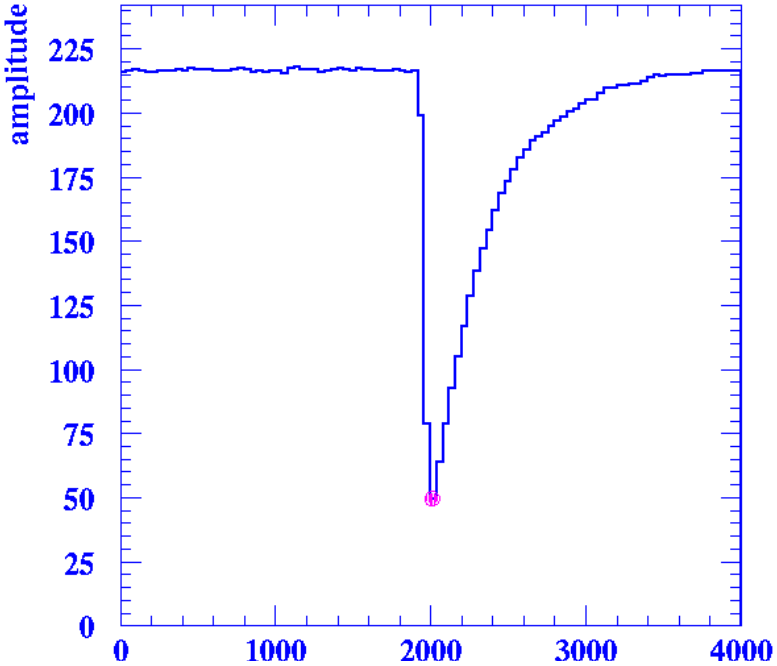


EVENT-LOOP


for all events
until end of run
TAPE
ReadEvent TPCio
Process Event?
CALL calcPedestal TPCcalcAmplitude
CALL subPedestal TPCcalcAmplitude
YES
Calibration?
Calibrate TPCcalcAmplitude
CALL calcAmplitudePad TPCcalcAmplitude
YES
Write DData?
DenseData TPCcalcAmplitude
CALL calcAmplitudeRow TPCcalcAmplitude
(Preselection)
ELSE
analyzEvent MYanalyzer
USER routine

analyzEvent

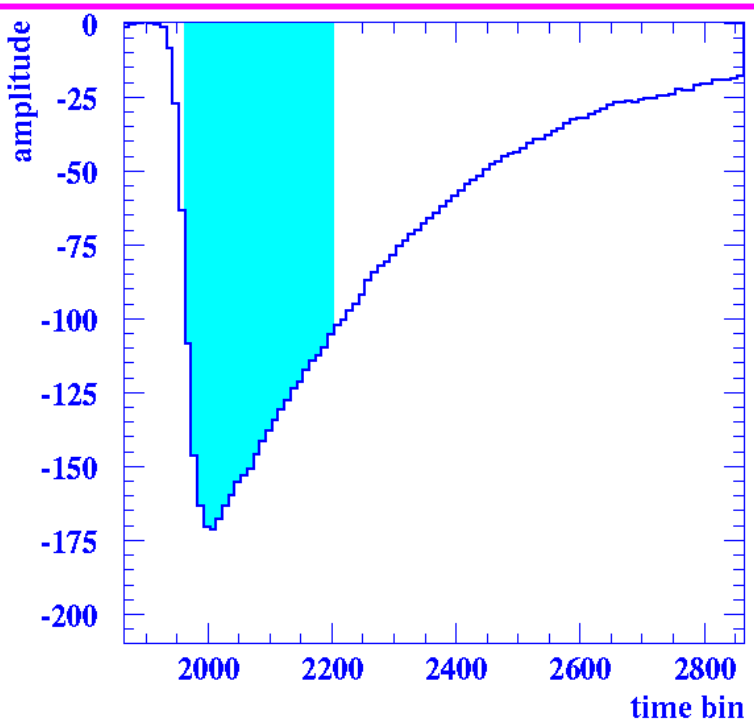




calcSignalFix

Amplitude = ADC(TMax)

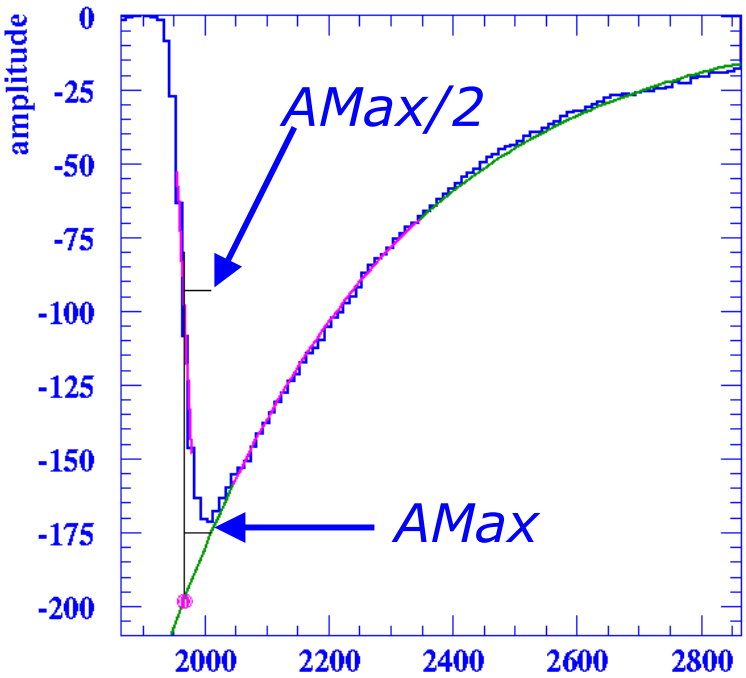
simple
great for debugging



calcSignalSum

Integral over ADC-Pedestal
from riseTime to fallTime

Intended for STAR pulses



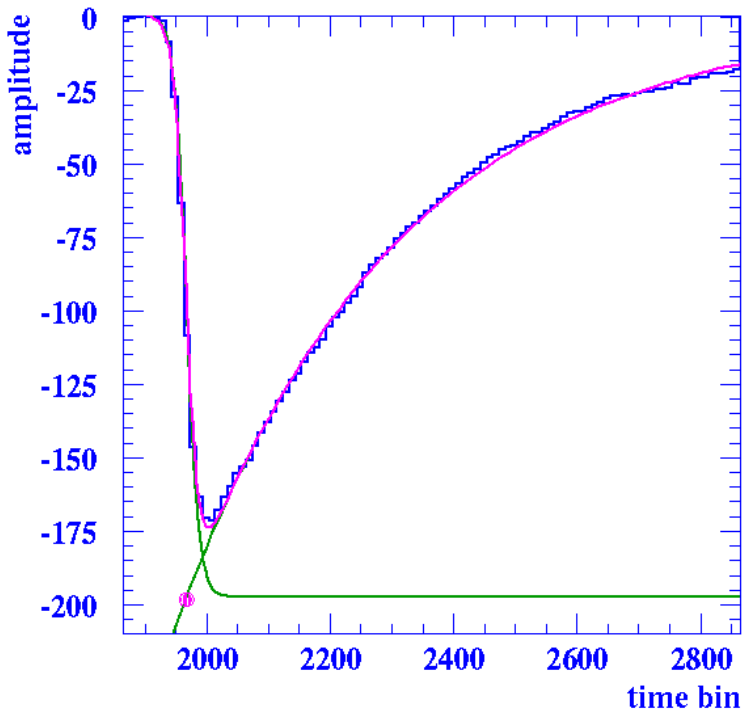
calcSignalExp

'fit' exponential with fallTime *fT*
*A(T) = AMax * exp((TMax-T)/fT)*

ADC(T0) = AMax/2

Amplitude = *A(T0)*

close to full fit



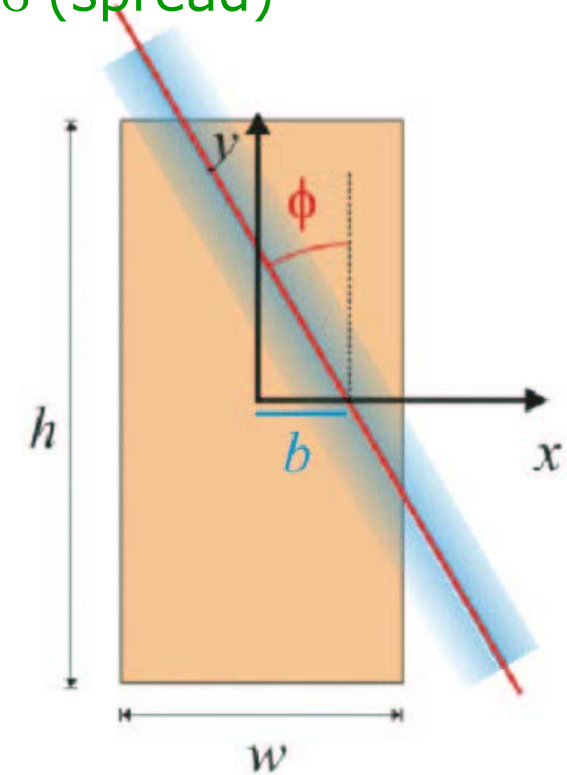
fitOnePulse

$$ped - \frac{amplitude}{1+\exp\left(\frac{-T+T0}{riseTime}\right)}\exp\left(\frac{-T+T0}{fallTime}\right)$$

Track Fit

Dean Karlens method

3 parameters:
x_0 (offset), ϕ (angle),
σ (spread)



TrackFitWidth — TPCtrack

fit all 3 free parameters

TrackFitXY — TPCtrack

width is fixed as function of drift distance

TrackFitX — TPCtrack

Only X_0 is free parameter

TRACKCHI — TPCtrack

calculate chisq:
intTrackPad gives expected charge
normalize expectation across row gives probability
Likelihood = Π probability * Amplitude

intTrackPad — Space

integral of track over pad
INTERFACE: code depends on pad_type

The Modules

DATA

TPCdata
all data information:
ADC spectra
Pads: location, amplitude, T0
Routines to retrieve information

Threshold
Information read in from TPCinput.txt

IOunits
IO unit numbers

RUN

TPCanalyzer
MAIN program

MYanalyzer
USER routines

RECO

TPCcalcAmplitude
ADC -> amplitude, T0

TPCtrack
Track reconstruction

Space
Point, line, track, rectangle
distance, integral, …

IO

TPCio
Reads input files

DataFormats
MIDAS, JTPC, DData

MyMinuit
wrapper for MINUIT

TPCpaw
keep hbook numbers local

TPCf77.f
F77 routines

Conclusion

Want to try something different?

Know FORTRAN better than JAVA?

Stuck with your test data and
need a simple package to analyze them?

GOTO
http://www.physics.carleton.ca/~sachs/TPC/F

Need more information - help to interface your data?

Email
kirsten.sachs@physics.carleton.ca