ARS simulation
in TriggerEfficiency (M. de Jong)

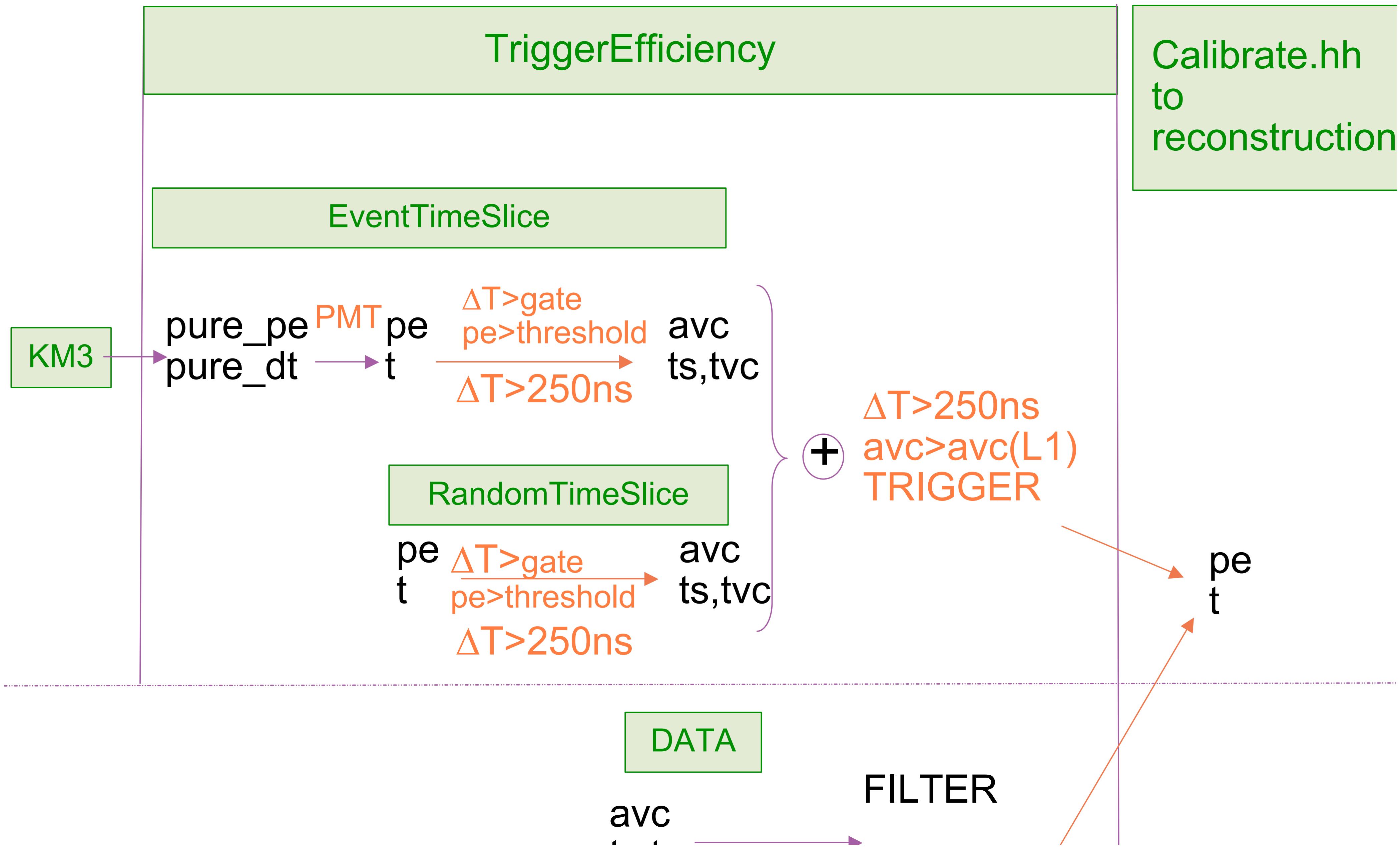
C.Donzaud (APC), A. Kouchner (APC), B. Vallage (IRFU)

I. What is done for timing and charge

II. What could be performed using in situ data concerning :
- dead time, charge & threshold of ARS
- optical background

Not discussed here :
- acceptance and efficiency of PMTs
- t0, tvc

General scheme



Part 1.

How to convert MC hits (photons on PMT) to SPE hits (time and charge) ?

PMT simulation (time & amplitude)

in EventTimeSlice

smearing : from pure_npe (on PMT) to npe
from pure_dt (on PMT) to t

t = Gaussian (pure_dt+t_event, TTS / sqrt(pure_npe))
pe = Gaussian (pure_npe*gain, gain_spread * sqrt(pure_npe))

TTS = 1.5 ns
gain_spread = 0.3 pe
gain=1

from PMT_interface.hh

OK !

Before, it was done in KM3

ARS integration time

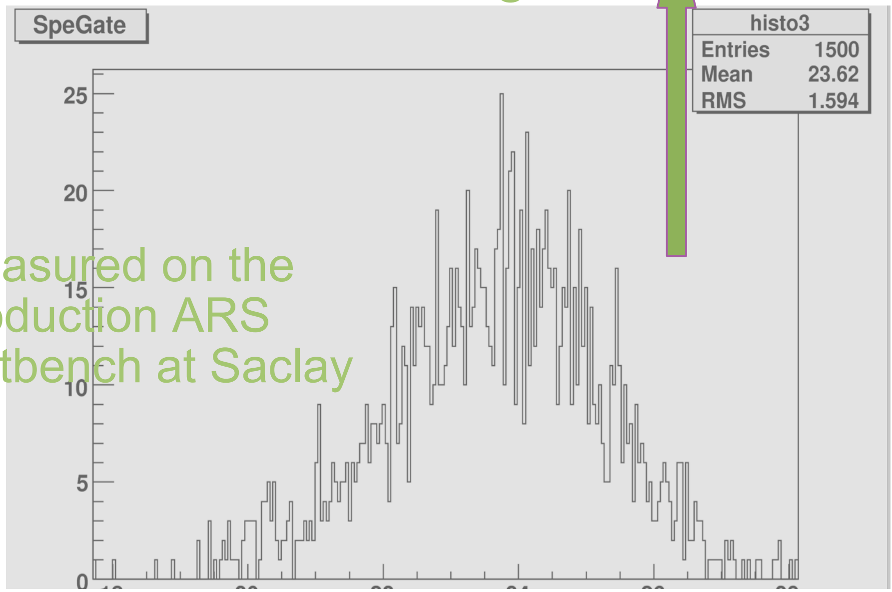
in EventTimeSlice and RandomTimeSlice independently

The token is given alternatively to the ARS partner

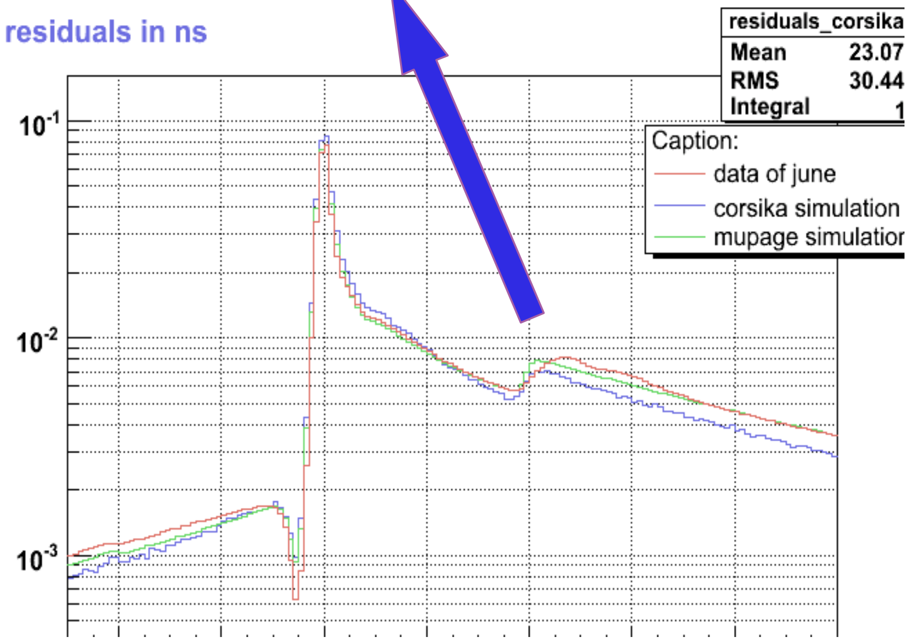
If (t(i+1,ARS0) – t(i,ARS1)) < gate) ⇒ pe(i)=pe(i+1)+pe(i)
with gate = 40 ns from ARS_**interface**.hh

gate = SPE_GATE + t_token_exchange
23.6 +/- 1.6 ns 16 ns

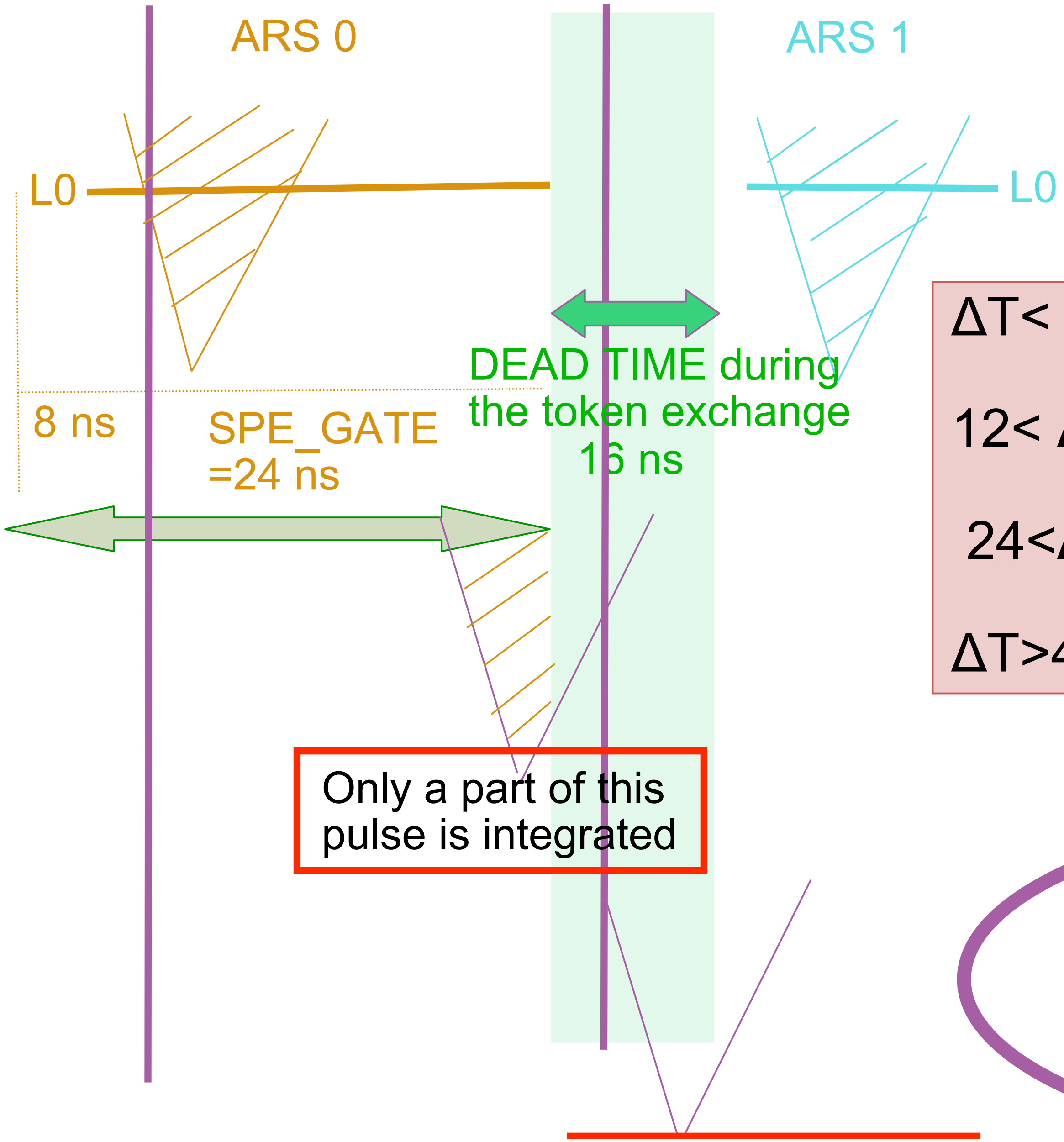
Integration window after the threshold crossing



Measured on the production ARS testbench at Saclay



ARS integration time



$\Delta T < 12$: pe(i)=pe(i)+pe(i+1)

$12 < \Delta T < 24$: pe(i)=pe(i)+pe(i+1)/α

$24 < \Delta T < 40$: hit(i+1) is lost

$\Delta T > 40$: hit(i+1) is integrated in ARS1

Proposition :

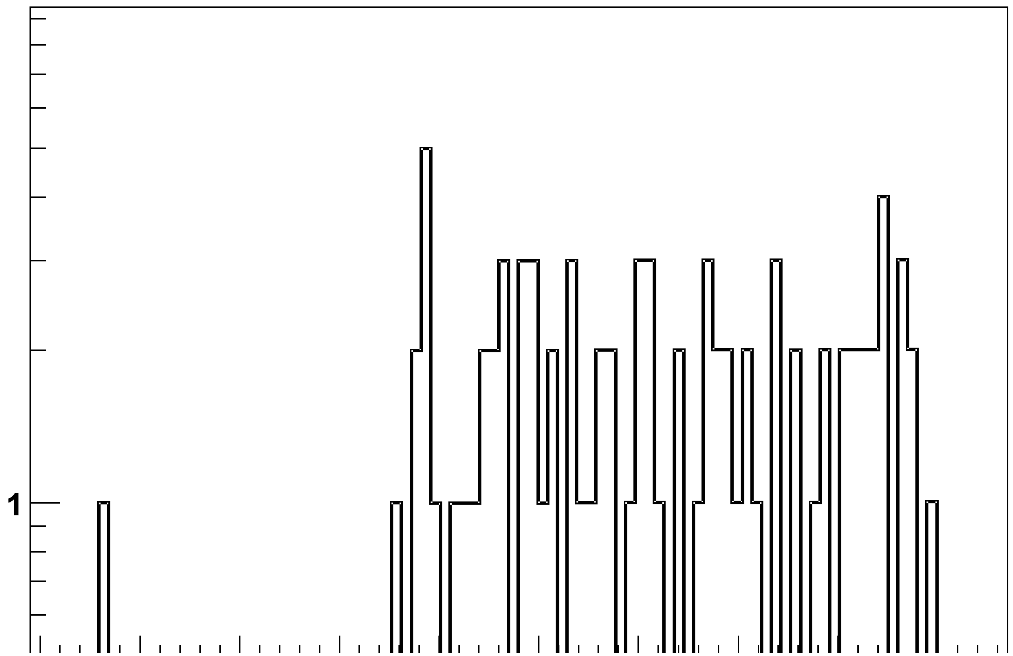
- **implement this effect**
- **no need to attribute value for individual ARS**

ARS dead time

1. inside a physics timeslice (in EventTimeSlice)
2. inside a random timeslice (in RandomTimeSlice)
3. after adding the background timeslice (in TriggerEfficiency.cc)

for a given ARS
if (t(i+1) - t(i)) < ARS_DEAD_TIME ⇨ hit(i+1) is removed

ARS_DEAD_TIME = 250 ns (defined in ARS_interface.hh)



ARS keeps the token when its partner is in dead-time (rate : 5×10^{-4})

GRB Run 32769

ARS L0 threshold

in EventTimeSlice

if(a_{hit} < threshold) ⟹ the hit is erased
threshold = 0.3 pe (defined in ARS_interface.hh)

Proposition :
Smear the threshold by firing in a Gaussian with
- mean = 0.3 pe
- sigma = 10% (1-2 DAC bin(s) of Trig0_th)
(1 bin * 1 mV/bin * 1/(45 mV) = 0.02 pe)
No need to attribute value for individual ARS ?

Slope of Trig0_th TF

1 pe = 45 mV

Conversion pe ⟷ AVC

method avc of SPE_writer

spe.avc = (int)(avc_1pe + (pe-1) / avc_slope)

avc_1pe = 21 (bits)
avc_slope = 0.08 pe/bit (=1/12.5)

} defined in the detector file

→ avc_0pe=8.5 (bits)

→ dynamic range = (255-8.5)*0.08=20 pe

method a of the avc calibration class (in reconstruction)

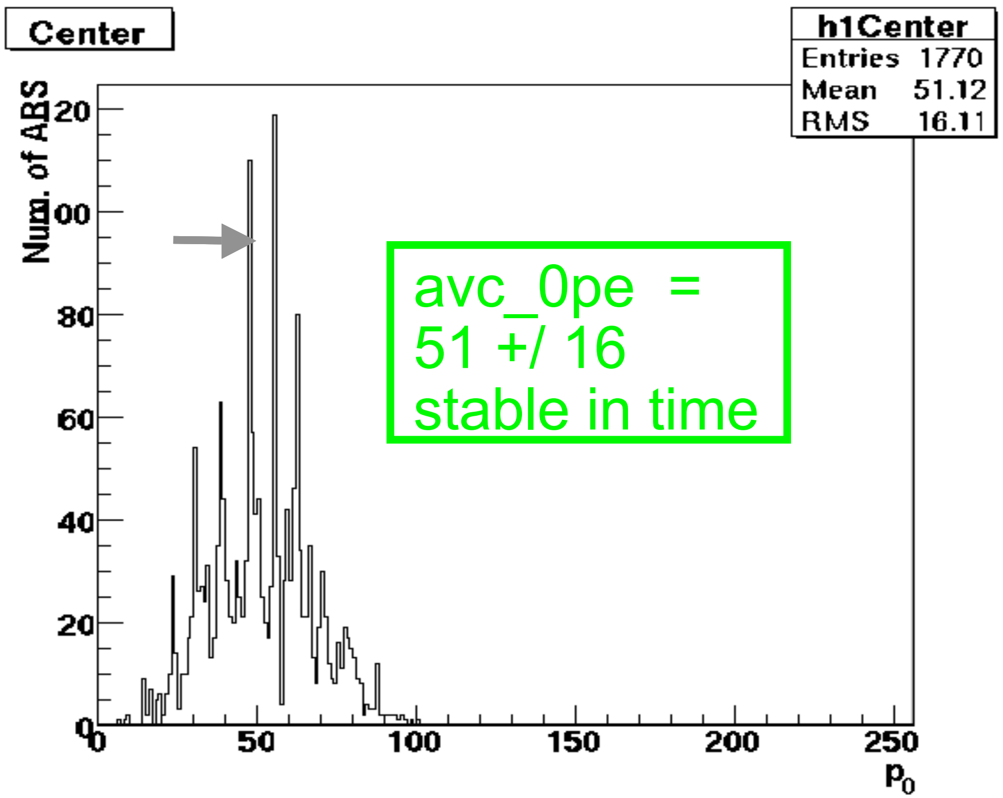
pe = 1 + avc_slope * (spe.avc - avc_1pe)

for high threshold L1 trigger

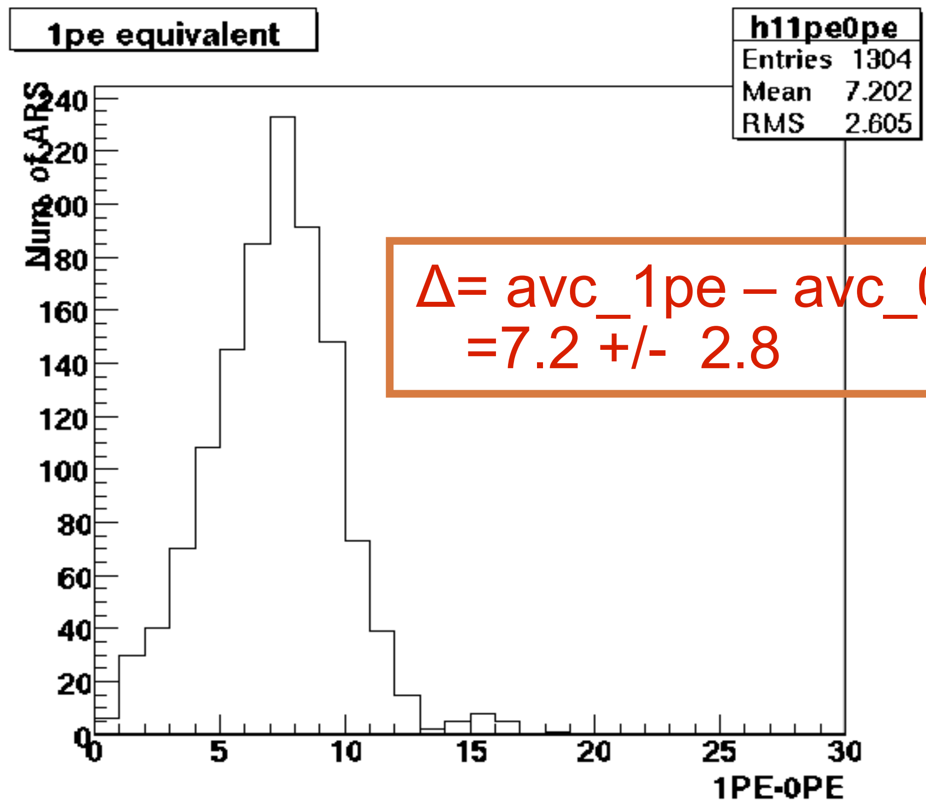
class trigger_L1 (called by TriggerEfficiency.cc)

L1 (pe) is converted into avc prior to be compared to the avc value of each hit.

In situ avc_0pe & dynamic range



avc_0pe = 51 +/ 16
stable in time



Δ= avc_1pe – avc_0pe
=7.2 +/- 2.8

Run 38470, Jan 09
see Bruny's talk

→ Mean dynamic range = (255-51)/Δ=26 +/- 10

Apply a more realistic calibration (I)

avc_0pe = 51 +/ 16
stable in time

Δ = avc_1pe – avc_0pe
=7.2 +/- 2.8

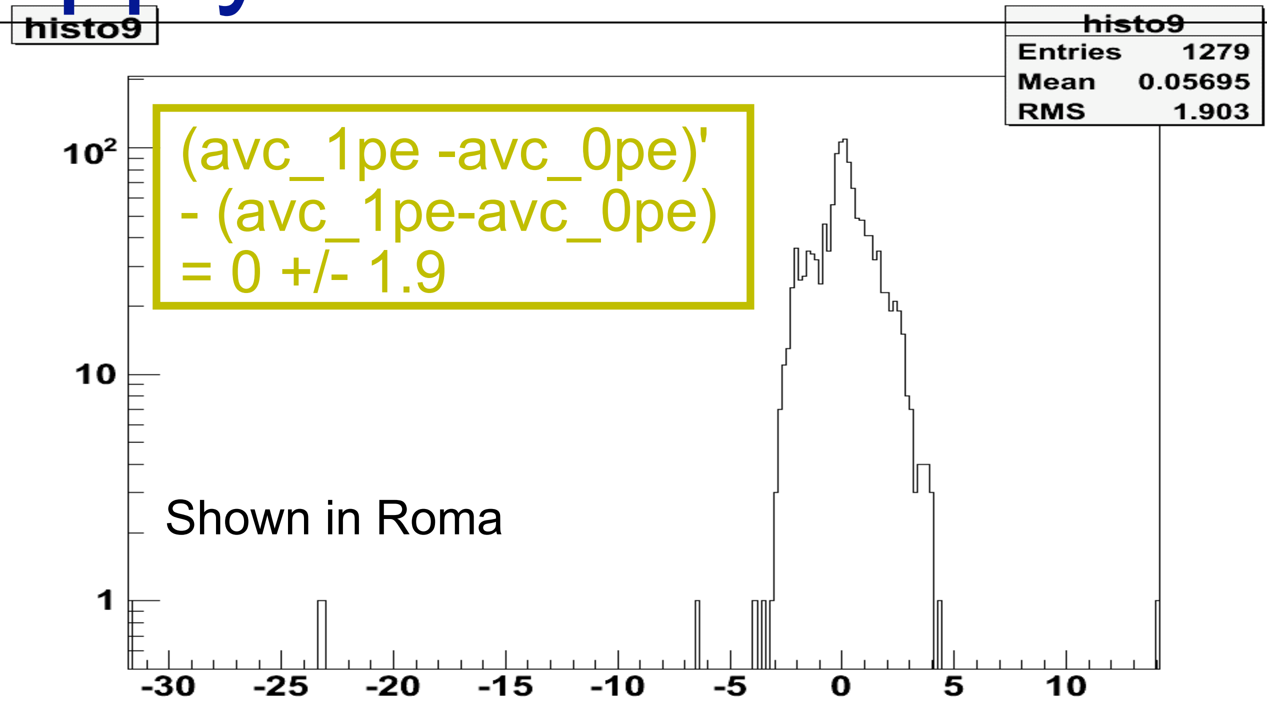
Proposition :

Table ARS_AVC_0pe_MC {LCM_id, ARSLINK, avc_0pe}
filled by firing avc_0pe in a Gaussian(<x>=51, σ=16)

Table ARS_AVC_1pe_MC {LCM_id, ARSLINK, avc_1pe}
filled by firing Δ in a Gaussian(<x>=7.2, σ=2.8) and
calculating avc_1pe **=** Δ + avc_0pe

spe.avc = (int)(avc_0pe **+ pe** * (avc_1pe - avc_0pe) **)**

Apply a more realistic calibration (II)



Same LSB but change of the base → Δ'-Δ=0 σ=DNL

To do : check that there is no correlation between Δ'-Δ and Δ

Proposition :

Table ARS_AVC_1pe_MC_smear {LCM_id, ARSLINK,avc_1pe'**}**
filled by firing avc_1pe' **in a Gaussian(<x>=** avc_1pe **, σ=1.9)**

pe = 1 + avc_slope *** (spe.avc –** avc_1pe'**)**
with avc_slope **= 1/(**avc_1pe' – avc_0pe**)**

Apply a more realistic calibration (III)

pe = f(avc) must be used :
- in TriggerEfficiency to compare pe to L1
- in avc_calibration class (called by Calibrate.hh) before the reconstruction

First : quantify the effect of this smearing
Later : take the dispersion of each ARS individually ?

avc_0pe is known to be stable in time but the decrease of avc_1pe with time implies to regularly update the new tables ?

ARS simulation (time)

class SPE_writer (called by EventTimeSlice)

spe.timestamp =f(t,   threshold,pe)
spe.tvc = f(t, tvc_min, tvc_max, threshold,pe)

Walk-effect defined in WalkCorrection

tvc0_min = 30 tvc0_max=225
tvc1_min = 30 tvc1_max=235

defined in the detector file

Not discussed here !

Part 2.

The optical background

PMT & ARS simulation for optical background

in RandomTimeSlice

$t_i = t_{i-1}$ + expdev()/mean_rate
- expdev: gives a number generated in an exp(-t) distribution
- mean_rate = nbOfHits_in_a_frame / frameDuration given by a timeslice fired at random from a run chosen by the user

Pe = TruncatedGaussian(gain, gain_spread,ARS_THRESHOLD)
- gain=1 pe
- gain_spread =0.3 pe (PMT)

from PMT_interface.hh

- ARS_THRESHOLD = 0.3 defined in RandomTimeSlice.hh

From hits to SPE_hits for optical background

In RandomTimeSlice

Removing of dead ARS's and PMTs for MC physics hits by erasing frames not present in background run : OK !

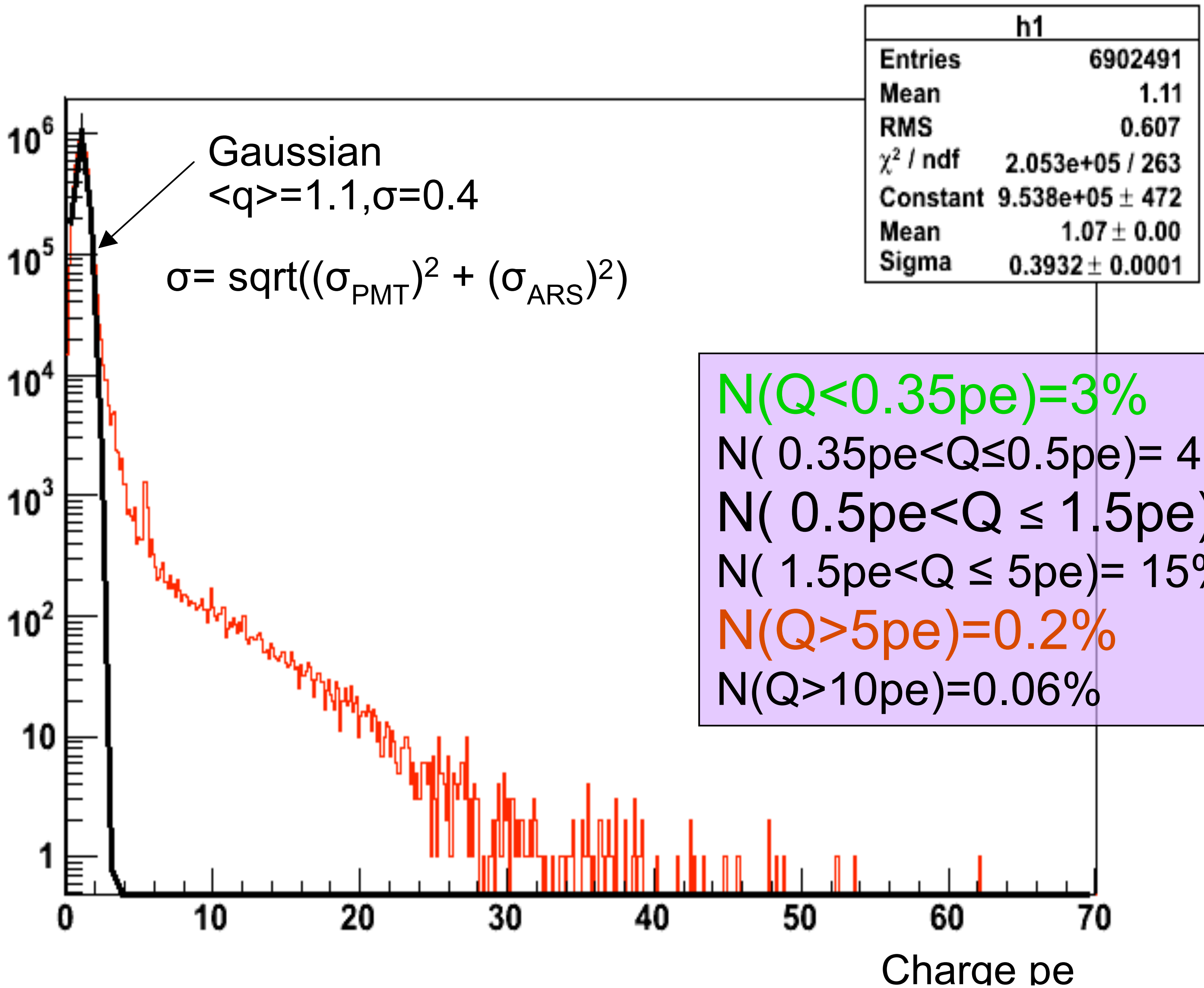
Same computation of TS, TVC and AVC (in SPE_writer) as for physics events except :

- no walk-effect (since q = 1 pe for all)

- if (spe.avc=avc(threshold)) spe.timestamp=0

shoud be implemented for physics hits in EventTimeSlice too

Measured charge distribution of background



N(Q<0.35pe)=3%

N(0.35pe<Q≤0.5pe)= 4%

N(0.5pe<Q ≤ 1.5pe)=78%

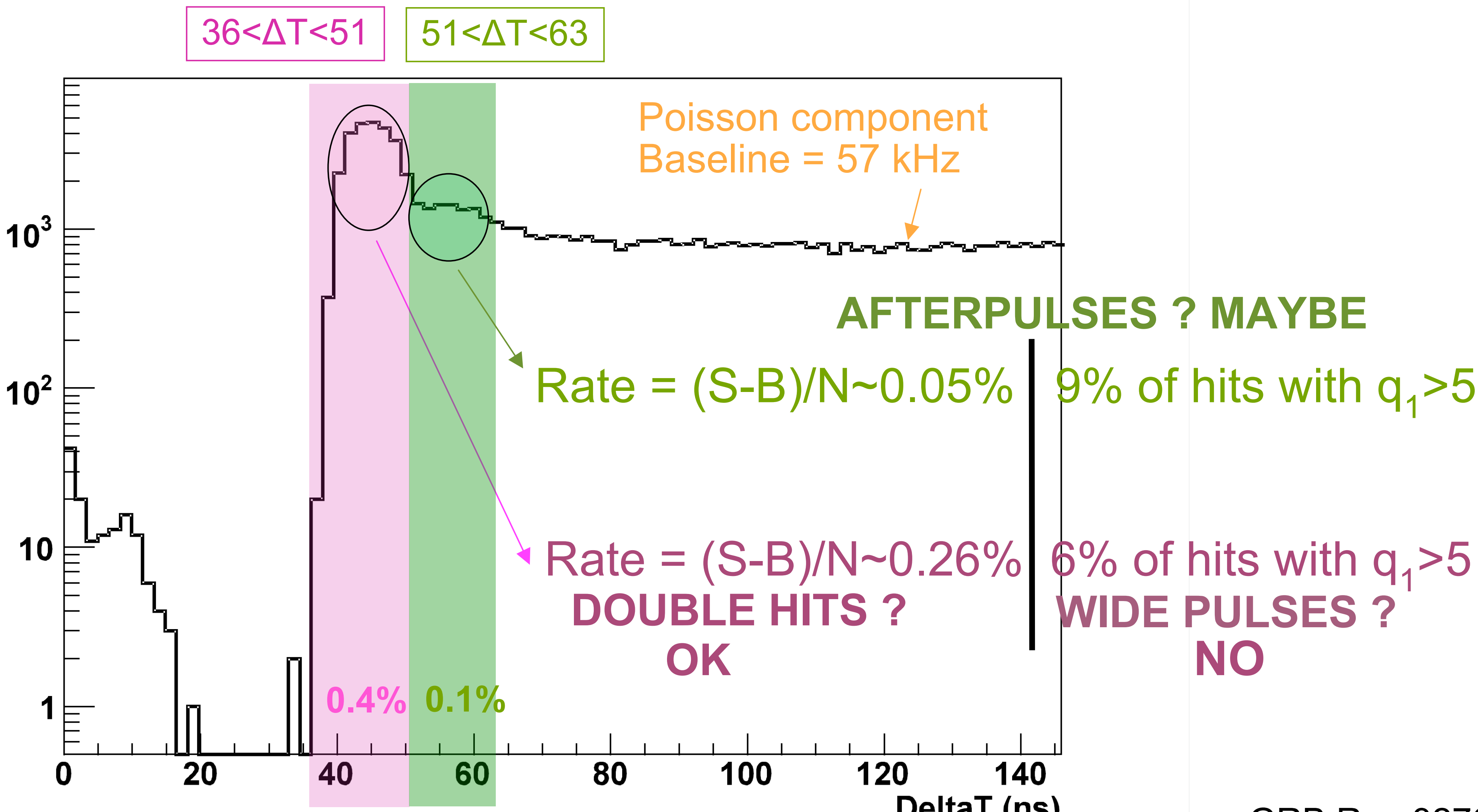
N(1.5pe<Q ≤ 5pe)= 15%

N(Q>5pe)=0.2%

N(Q>10pe)=0.06%

GRB Run 32769
Shown in Sinaia

Measured DeltaT (between 2 consecutive hits in PMT) distribution



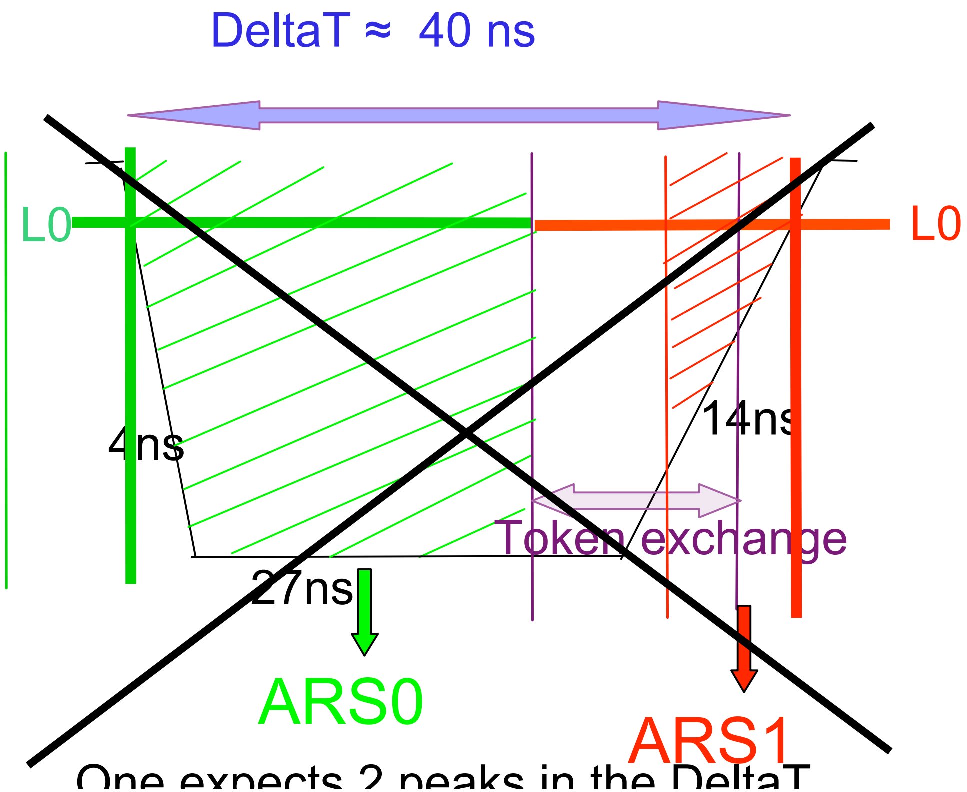
GRB Run 32769
Shown in Sinaia

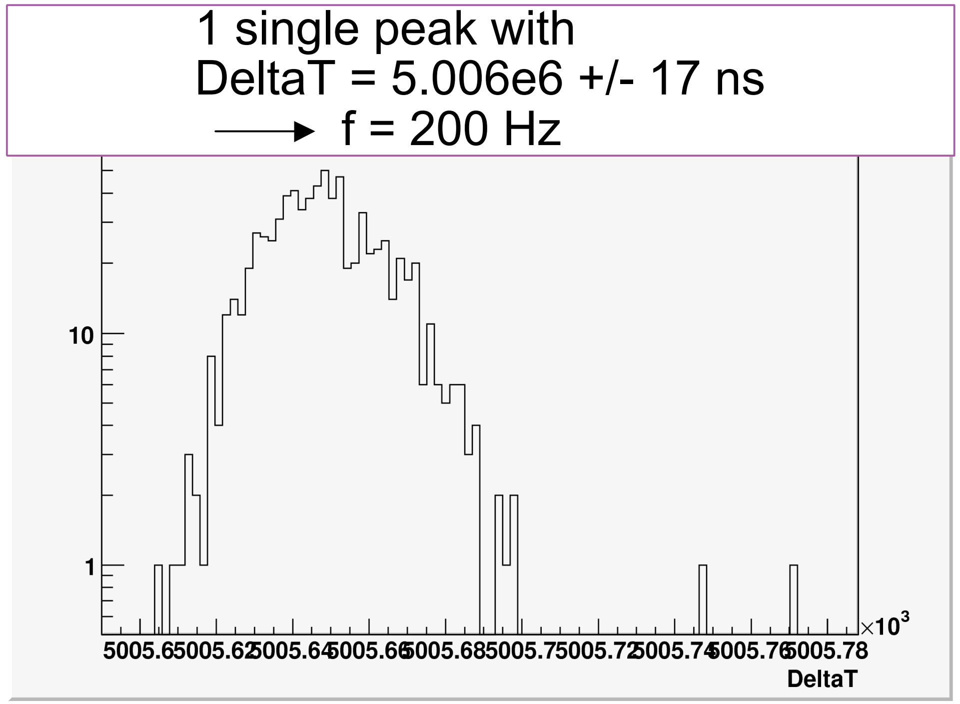
Wide pulses

We thought that the peak around 43 ns could come from wide pulses : FALSE

ARS expert : "The ARS triggers only when the signal goes below the L0 level and cross it on rising edge"

This is confirmed by a measurement made on Saclay DAQ testbench with a generator frequency = 200 Hz and wide pulses :

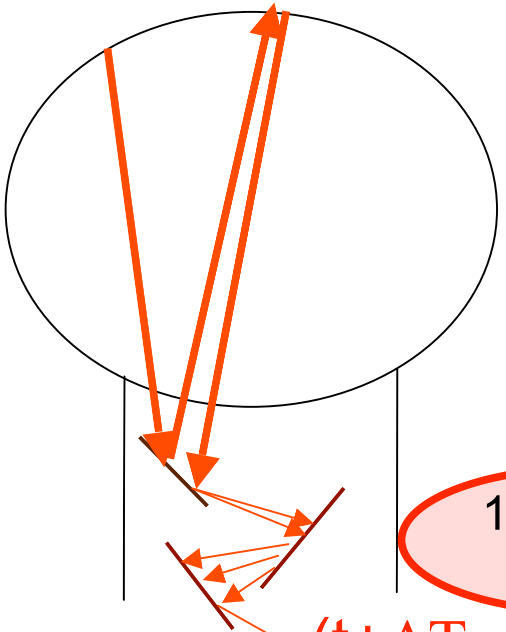




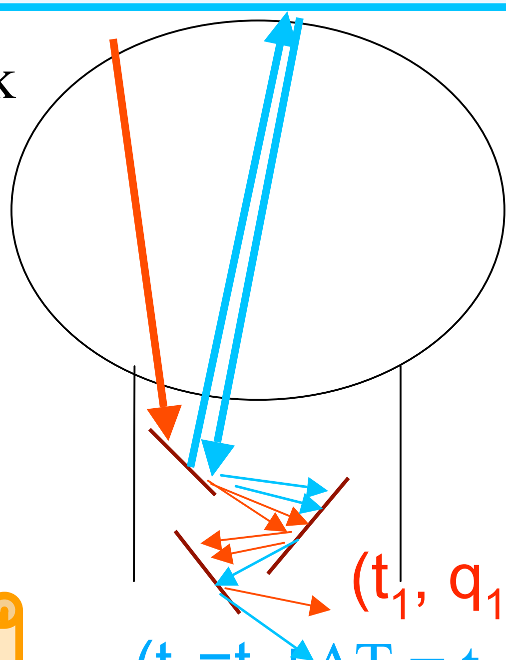
Afterpulses type 1 & Delayed pulses (I)

Delayed pulses : **one** e- coming from the photocathode makes an elastic collision on the first dynode (D1) or potential grids, returns towards the photocathode (PK)

Afterpulse type 1 : **one photon** emitted from the first dynode (D1) or potential returns towards the photocathode (PK)



In both cases, the e- collides back onto the PK and is focused to the first dynode once more



1 delayed hit

$(t+\Delta T = t + 2t_{PK\text{-}D1}\ ,\ q=1\ \text{pe})$

but if N>1 photons on PK,

2 hits

1 hit (t,q= N-1 pe)

1 hit (t+ΔT,1pe)

The probability increases with the photon number on PK

$(t_1,\ q_1)$ 1 hit

$(t_2=t_1+\Delta T = t_1+ 2t_{PK\text{-}D1}\ ,\ q_2=1\text{pe})$

1 delayed

Afterpulses type 1
& Delayed pulses (II)

MEASUREMENTS ON TESTBENCH

at Saclay on 912 PMTs with an oscilloscope, in [t_1+10 : t_1 +100] ns

(NIMA 555 (2005) 132-141)

Rate Afterpulses type 1 : 1.5 ± 0.3 %

Rate Delayed pulses : 3.7 ± 0.2 %

5 %

IN SITU MEASUREMENTS

- same time window : [36-63] ns
- increase of the probability with the charge

but

- rate < 0.3 %.

Why ?

- decrease of large charges in sea (^{40}K of the glass sphere makes larger charge then bio-luminescence)
- the ARS integration window cut the second pulse ? (To answer : analyse a dark room run)

Use more realistic background ?

Proposition :
Generation of couples (dt,pe)
with dt and pe generated from a
measured distribution (DeltaT,q_1)

- Physics hits (from KM3) have larger charge (charge distribution ?)
- Is it necessary to simulate the 2 peaks between 36 and 63 ns for physics hits ?

Remarks about adding the background

1. We loose the token ring behaviour when adding the optical background : the order list ARS1 ARS0 ARS0 ARS1 is possible in MC.
2. We don't add the amplitude of a background hit and a physics hit when their DeltaT < gate.

It would be more realistic to merge the 2 lists of hits (physics and background) before to simulate the ARS. Need to quantify the effect before doing so !

Summary

Some propositions to perform the ARS simulation

- Implement the dead time more carefully (no need to have value for individual ARS)

- Smear the threshold of 10% (no need to have value for individual ARS)

- Apply a more realistic charge calibration :

    - More realistic AVC dynamic range

    - Smear the avc_1pe value

  Need of 3 new tables with values
  - 1. fired in in-situ distributions
  - 2. given by each individual ARS

- Optical background : generation of couples (dt,pe) with dt and pe generated from a measured distribution (DeltaT,q_1)

- Apply the ARS dead-time after the adding of the optical background at the KM3 hits

Need to define procedures to evaluate the contribution of each modification ?

END

Where are the files ?

Root : antares-daq

tools/MonteCarlo/TriggerEfficiency.cc
tools/TimeSlice/EventTimeSlice.hh called by TriggerEfficiency.cc
tools/TimeSlice/RandomTimeSlice.hh called by TriggerEfficiency.cc

tools/TimeSlice/PMT_interface.hh
tools /TimeSlice/ARS_interface.hh

include/trigger/WalkCorrection.hh
include/trigger/trigger_L1.hh, called by TriggerEfficiency.cc

include/trigger/trigger_io.hh
SPE_writer : defined in antares-daq/include/trigger/spe_io.hh

to convert pe to avc : method avc of the class SPE_reader defined in trigger_io.hh.
To convert a pe of one hit to avc, the operator () of the class SPE_writer is used. This operator calls the avc method of SPE_reader.

To convert avc to pe :
method a of the avc calibration class which is defined in trigger_io.hh. The methode a is called by
Calibrate_TimeSlice which is defined in
$ANTARES/src/Physics/v2r1/inc/Calibrate.hh

Calibration values in detector files

avc_1pe = 21
avc_slope = 0.08 (dynamic range = 20 pe)
tvc0_min = 30 tvc0_max=225
tvc1_min = 30 tvc1_max=235

are defined with the key : DefaultSPEreader in the detector files and read by SetUserPreferences(DefaultARSparameters) in TriggerEfficiency :

d10_c00_s01.det: 0 21 0.047 0 255 0 255 old ?
d10_c00_s01.params:: 0 21 0.047 0 255 0 255 old ?
d10_c00_s02.det: 0 21 0.08 30 225 30 235 ok
d10_c00_s02.params: 0 21 0.08 30 225 30 235 ok
l05_c00_s00.det: 0 21 0.08 30 225 30 235 ok
l05_c09_s01.det: 0 21 0.047 0 255 0 255 old ?
l05_c09_s02.det: 0 21 0.08 30 225 30 235 ok
l12_c00_s00.det: 0 21 0.08 30 225 30 235 ok

How to defined PMT & ARS parameters and Calibration values for individual ARS

PMT parameters
TTS, gain, gain_spread (now defined in PMT_interface.hh)
use addPMTparameters of PMT_interface.hh

ARS parameters
threshold, gate (now defined in ARS_interface.hh)
use addARSparameters of ARS_interface.hh

Calibration values (now defined in detector file)
t_offset avc_1pe avc_slope tvc0_min tvc0_max tvc1_min tvc1_max
use set method of TriggerInterfaceModel defined in trigger_io.hh

Remarks about adding of background

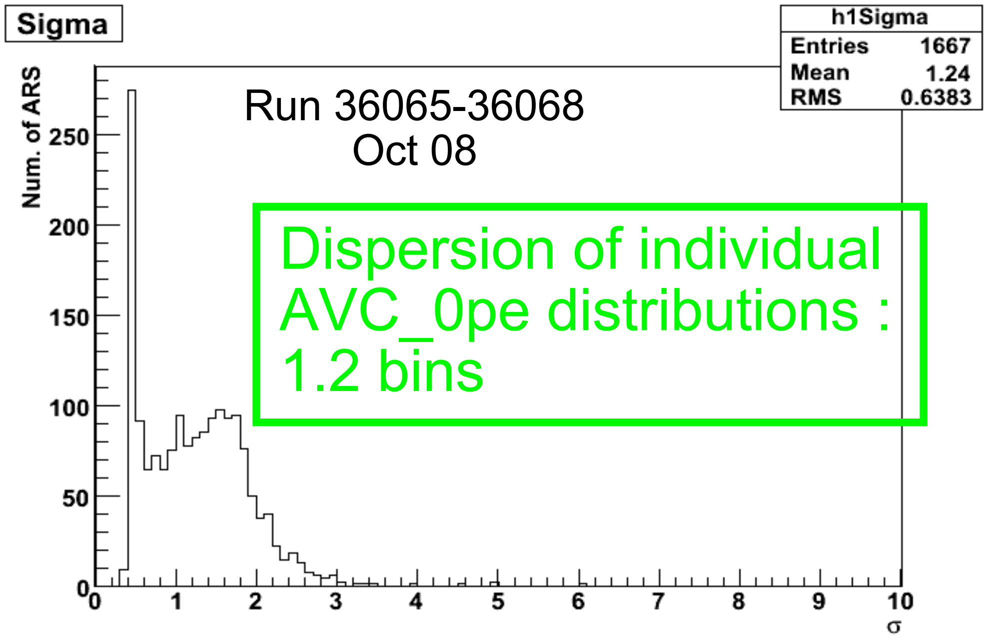
Question posée à Maarten :
6. We can have the arrangement
... ARS1 ARS0 ARS0 ARS1 ... once we add physics hit and random hits
7. We don't add amplitude of a background hit and a physics hit when they have DeltaT < gate. So we can have values smaller than gate in DeltaT MC distribution.

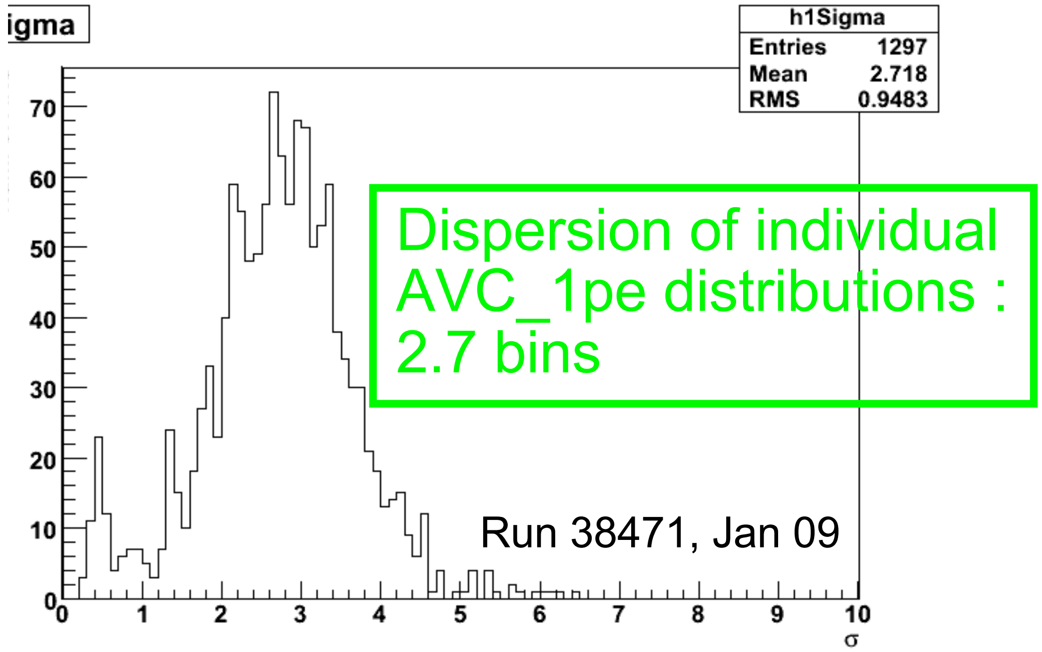
Reponse of Marteen :
The token ring behaviour is not retained -as you stated- when adding (random) background.
This is mainly due to maintain the possibility to add a SPE_TimeSlice (e.g. from a GRB) to the Monte Carlo event.
The same (see 6.) is true for the amplitude.
Points 6. and 7. could be pursued, but I hope someone can quantify the effect before.

New interface of Maarten

Marteen is presently defining the general interfaces for the ARS and PMT parameters.
He assumes that some day, we want to specify different values for the ARSs and PMTs in the detector.
The syntax for this will look like
  PMT <LCM> <position> TTS=<value>, QE=<value>;
  PMT <LCM> <position> TTS=<value>, QE=<value>;
global parameter values can be set as follows
  %.QE=>value>
And idem dito for the ARS parameters.

Measured avc_0pe & avc_1pe





Rq d'Antoine : dans cette distrib, le cross talk n'est pas corrigé

Rq d'Antoine : dans cette distribution il y a les 0.3 pe du aux PM qui ont été mis dans la simulation avant lors du passage pure_pe à pe

avc_slope=(0-1)/(avc_0pe-avc_1pe)

$(\sigma_{slope})^2= 1/(avc\_0pe-avc\_1pe)^4 \; ((\sigma_{avc\_0pe})^2+(\sigma_{avc\_1pe})^2)$

$\sigma_{slope}=0.018$

RandomTimeSlice

pe=TruncatedGaussian(gain,gain_spread,ARS_THRESHOLD)

- gain=1 pe
- gain_spread =0.3 pe from PMT_interface.hh

- ARS_THRESHOLD = 0.3 in RandomTimeSlice.hh.

Question at Marteen : Why we don't use the threshold parameter from ARS_parameter.hh instead of this one (which would allow to smear it as in EventTimeSlice) ??

Reponse de Marteen :

1. The ARS_THRESHOLD = 0.3 has been defined to make sure that the observed singles rate is retained when the threshold is set at 0.3 p.e.
   A higher threshold (from ARS_interface.hh) will thus lead to a lower count rate, as it should.

J'ai juste l'impression qu'on calcule pour rien ???

2. I looked at the various constructors of the RandomTimeSlice class. Indeed -as you stated- the constructor that uses the summary data should also use the ARS threshold specified in ARS_InterfaceModel. The other constructor should -to my understanding- use the common threshold ARS_THRESHOLD to interprete the rate correctly for the reason I explained earlier.

Proposition : Generation of couples $(t_{i-1}+dt_i,pe_i)$
with dt_i and pe_i generated from a measured
distribution (DeltaT,q_previous)
with the TunuranDiscrDist method of root
in RandomTimeSlice.hh