```
SetDirectory["~bastian/Desktop/1800txt"];
Namelist = FileNames[]
Namelist // Length
nfiles = %
```

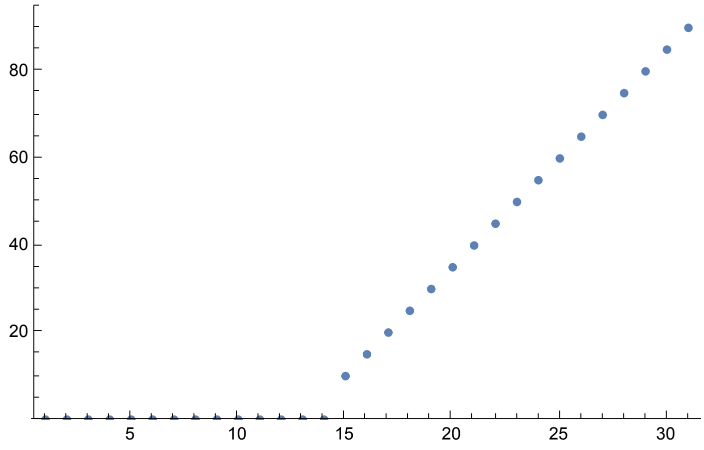
```
{.DS_Store,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-100um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-105um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-10um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-110um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-1.1um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-120um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-125um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-130um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-135um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-140um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-145um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-150um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-155um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-15um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-160um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-165um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-170um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-20um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-25um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-2um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-30um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-35um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-40um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-45um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-50um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-55um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-5um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-60um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-65um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-70um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-75um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-80um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-85um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-90um_tr.txt,
 Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-95um_tr.txt,
 simlandau1800.dat, updateBichsel.txt, xcelfromlu.csv}
```

```
39
```

```
39
```

```
Clear[de];
de = ConstantArray[0, 33];
nfil = {3, 14};
AppendTo[nfil, Range[18, 33]];
nfil = Flatten[nfil];
filename = Namelist[[3 + 1]];
Print[filename];
a = ToExpression[StringDrop[StringDrop[filename, 55], -9]];
de[[15]] = a;
filename = Namelist[[14 + 1]];
Print[filename];
a = ToExpression[StringDrop[StringDrop[filename, 55], -9]];
de[[16]] = a;
Do[
  filename = Namelist[[i + 2]];
  Print[filename];
a = ToExpression[StringDrop[StringDrop[filename, 55], -9]];
  de[[i]] = a;
, {i, 17, 33}];
de = Drop[de, {26}];
de = Drop[de, {19}];
ListPlot[de, PlotRange → {Full, {0, 95}}]
```

```
Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-10um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-15um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-20um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-25um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-2um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-30um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-35um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-40um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-45um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-50um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-55um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-5um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-60um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-65um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-70um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-75um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-80um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-85um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-90um_tr.txt
```




```
d = Drop[de, 14]
```

{10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90}

```
nfil = Drop[(nfil), {12}];
nfil = Drop[(nfil), {5}]
Dimensions[nfil]
```

{16}

```
Do[
  filename = Namelist[[nfil[[i]] + 1]];
  Print[filename];
  data = Import[filename, "Table"];
  scopedata = Cases[data, {_?NumberQ, _?NumberQ}];
  scope[[i]] = scopedata;
  (*{time[[3]],v[[3]]}=scopedata*)
  , {i, 1, 16}];
```

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-10um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-15um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-20um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-25um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-30um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-35um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-40um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-45um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-50um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-55um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-60um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-65um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-70um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-75um_tr.txt

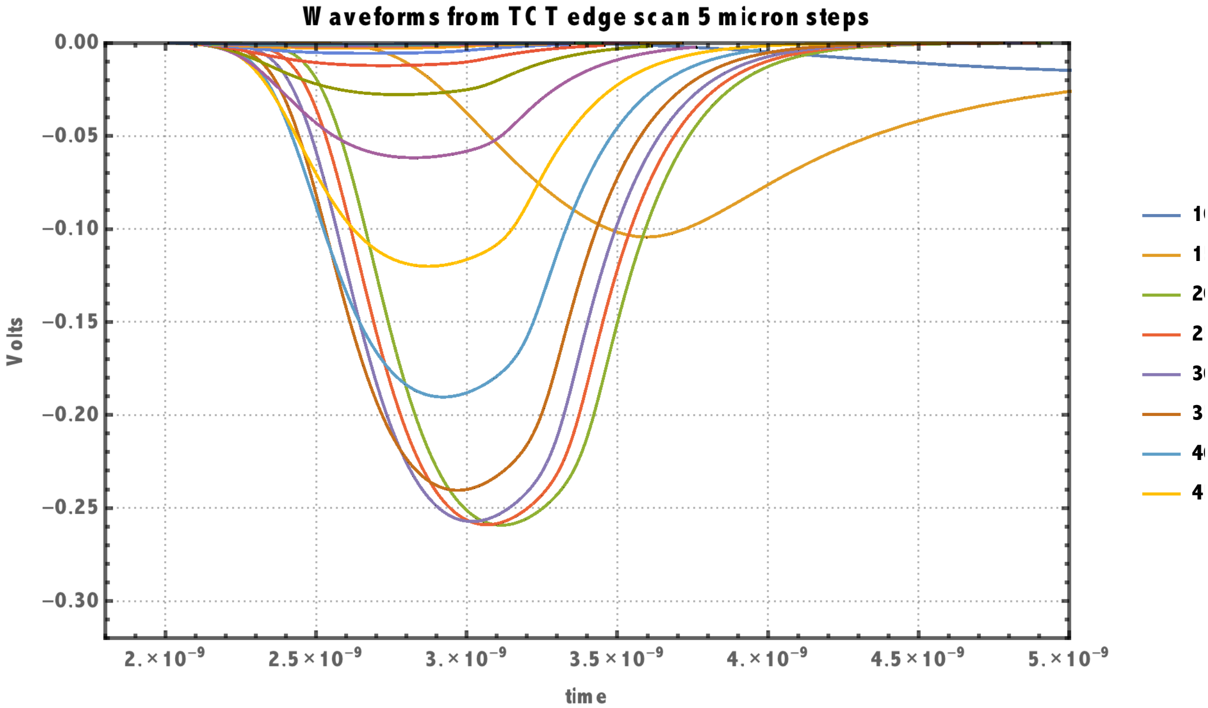
Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-80um_tr.txt

Fluence-0-tct-1060nm-APD-1.4e14-200um-junct-57um-1800V-85um_tr.txt

```
Do[
 ddd = Dimensions[scope[[i]]][[1]];
 Print[i, "     ", ddd];
 , {i, 16}]
```

```
1    188
2    336
3    2163
4    2034
5    1988
6    2105
7    1388
8    2041
9    1255
10    1203
11    1096
12    1095
13    1320
14    1012
15    794
16    639
```

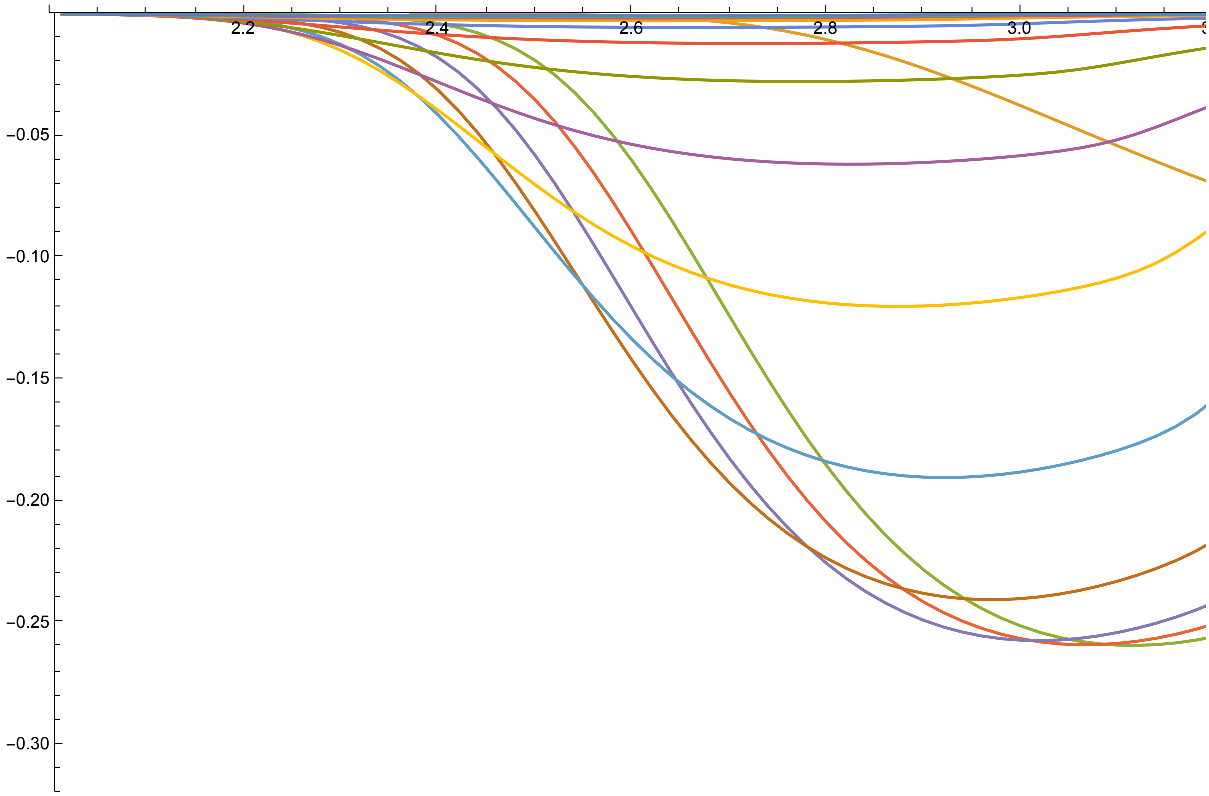
```
ListPlot[{scope⟦1⟧, scope⟦2⟧, scope⟦3⟧, scope⟦4⟧, scope⟦5⟧, scope⟦6⟧,
  scope⟦7⟧, scope⟦8⟧, scope⟦9⟧, scope⟦10⟧, scope⟦11⟧, scope⟦12⟧,
  scope⟦13⟧, scope⟦14⟧, scope⟦15⟧, scope⟦16⟧}, PlotTheme → "Detailed",
 PlotRange → {{0.18/10^8, 0.5/10^8}, {16 * (-0.02), 0.00}}, Joined → True,
 FrameLabel → {{HoldForm[Volts], None}, {HoldForm[time], None}},
 PlotLabel → HoldForm[Waveforms from TCT edge scan 5 micron steps],
 LabelStyle → {FontFamily → "Abadi MT Condensed Extra Bold", 12, GrayLevel[0]},
 ImageSize → Large, Frame → True,
 FrameStyle → Directive[GrayLevel[0, 0.62], Thickness[Large]], PlotLegends → d]
```



Save instead as digital samples with 20 picosecond sampling frequency. This deals with the problem of not being able to do simple operations
with Interpolation functions (ie like adding them to make a new function). The end result also looks like scope data so it will be easy to analyze.

```
edgescan = ConstantArray[0, {16, 65}];
Do[
  nent = Dimensions[scope[[i]]][[1]];
  tt = Table[scope[[i, j, 1]], {j, 1, nent}] * 10^9;
  vv = Table[scope[[i, j, 2]], {j, 1, nent}];
f = Interpolation[Transpose[{tt, vv}]];
  edgescan[[i]] = Table[f[1.98 + j * 0.02], {j, 1, 65}];
  , {i, 1, 16}];
time = Range[65] * 0.02 + 1.98;
```

```
waveplot = Table[Transpose[{time, edgescan[[i]]}], {i, 1, 16}];
ListPlot[waveplot, Joined → True, PlotRange → {{2, 3.3}, {16 * (-.02), 0.}}]
```



A number of Landau Fluctuation forms have been used to introduce weighting of the slices :

1) An approximate "Landau pdf" with parameters only guessed to give agreement with measured sum waveforms

2) A "Landau pdf" with parameters to best fit the Bichsel form for 5 micron Silicon

3) A new pdf constructed directly from Su Dong' s calculation with the Bichsel software

```
fullDat = ReplacePart[Import[
    "~bastian/Desktop/Bichsel_spectrum_Muon_1GeV_5micron.dat"], 1 → {0, 0, 0}];
e = fullDat[[All, 1]];
prob = fullDat[[All, 2]];
ListPlot[Transpose[{e / 1000., prob (3000 / 0.8)}]]
```

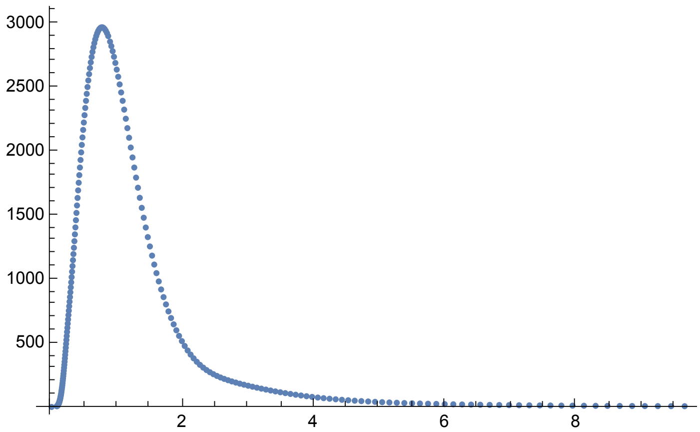


Figure 1

```
f = Interpolation[Transpose[{e / 1000., prob}], InterpolationOrder → 1];
 𝒟 = ProbabilityDistribution[f[x], {x, 0, 6}, Method → "Normalize"];
 PDF[𝒟, x];
RandomVariate[𝒟, 20 000];
 Show[
  {Histogram[%, {0, 10, .2}], ListPlot[Transpose[{e / 1000., prob (3000 / 0.8)}]]},
  AxesLabel → {HoldForm[Energy loss per 5 micron in keV], HoldForm[frequecy]},
  PlotLabel → HoldForm[Overlay of Su Dong′ s calculation and output
     of my random number generator], LabelStyle → {GrayLevel[0]}]
```

```
Show[%118,
 AxesLabel → {HoldForm[Energy loss per 5 micron in keV], HoldForm[frequecy]},
 PlotLabel → HoldForm[Overlay of Su Dong′ s calculation and output
    of my random number generator], LabelStyle → {GrayLevel[0]}]
```

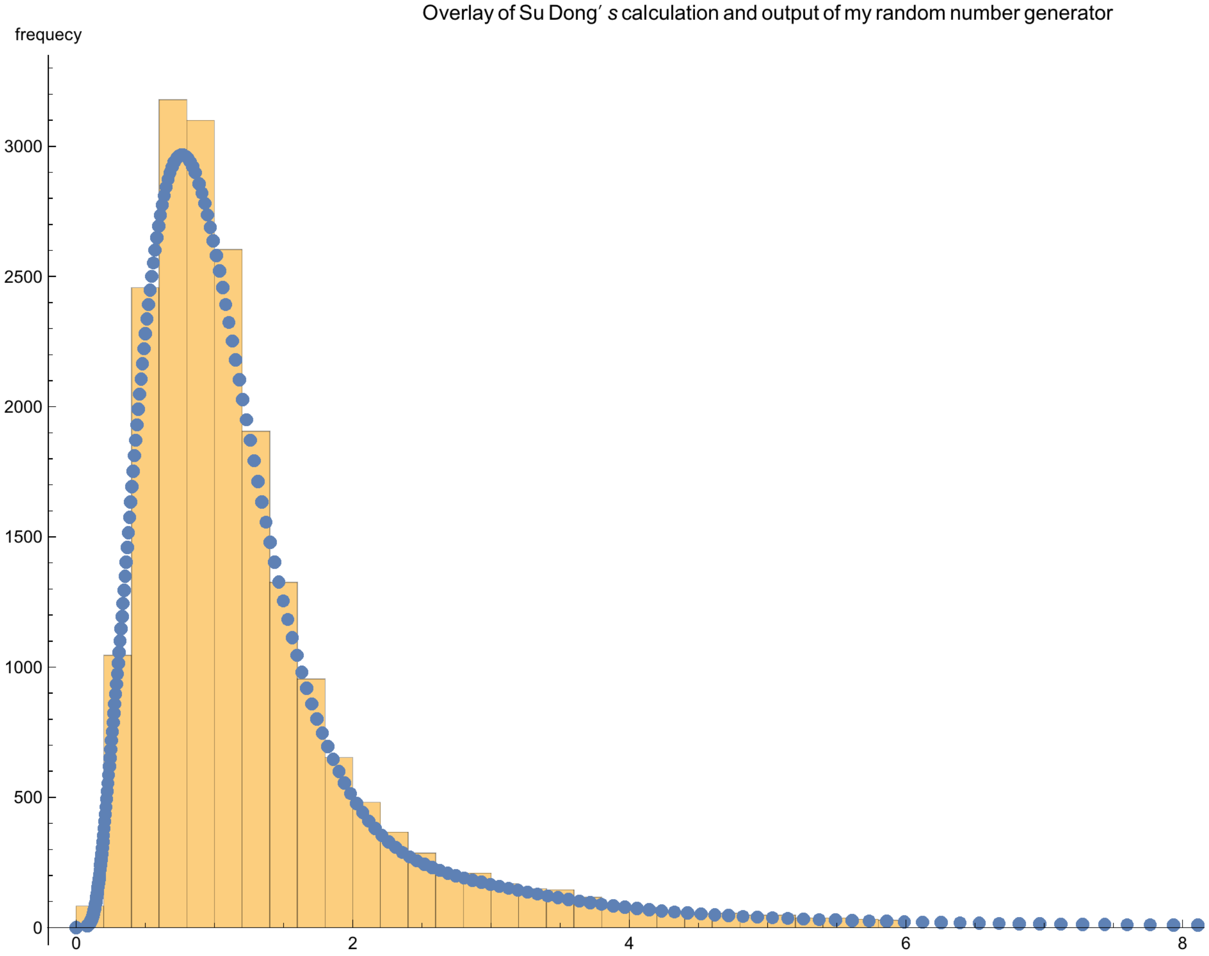


Figure 2

```
(*σ=0.08;μ=1.2;*)
(*σ=0.25;μ=1.2;*)
σ = 0.2; μ = 0.68;
data = RandomReal[LandauDistribution[μ, σ], 10^6];
Show[
 {Histogram[data, {0.0, 4, 0.1}, "PDF", AxesOrigin → {0.0, 0}],
  Plot[PDF[LandauDistribution[μ, σ], x], {x, 0.0, 4},
   PlotRange → Full, PlotStyle → Thick, ImageSize → Large],
  ListPlot[Transpose[{e / 2000., prob (1.4 / 0.8)}]]},
 AxesLabel → {HoldForm[1/2 Energy loss in keV], HoldForm[frequency]}, PlotLabel →
  HoldForm[Overlay of Landau Approximation to Su Dong and SuDong′ s points],
 LabelStyle → {GrayLevel[0]}]
```

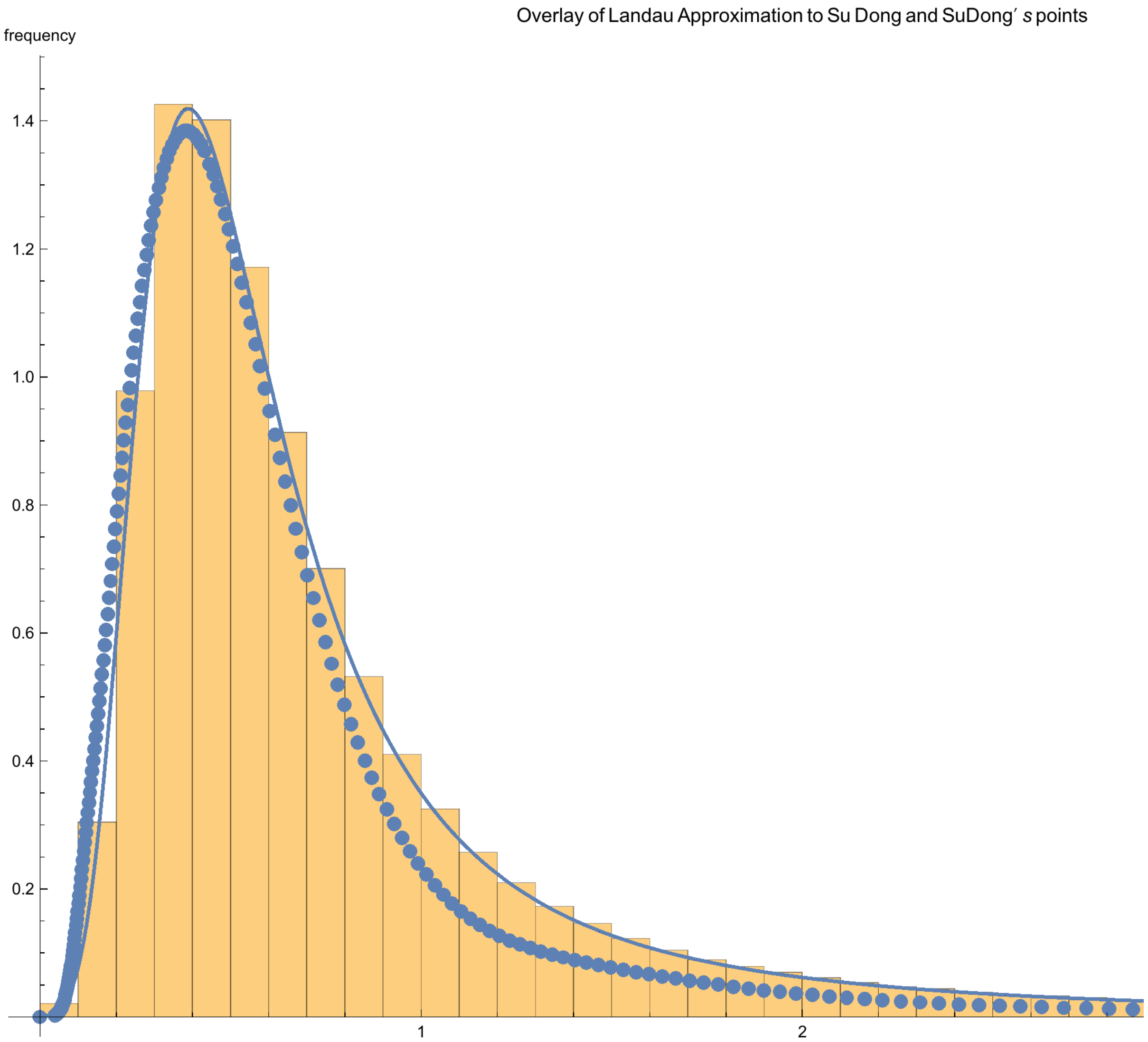


Figure 3

This is an earlier approximate Landau Distribution which gave small (~20 picosec) time jitter with CF algorithm.

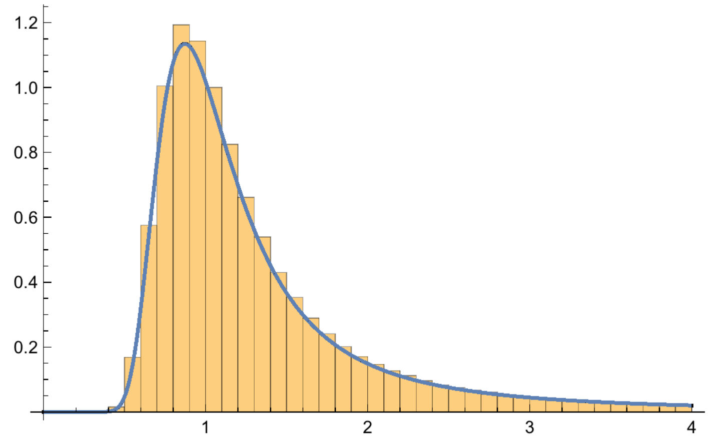


Figure 4

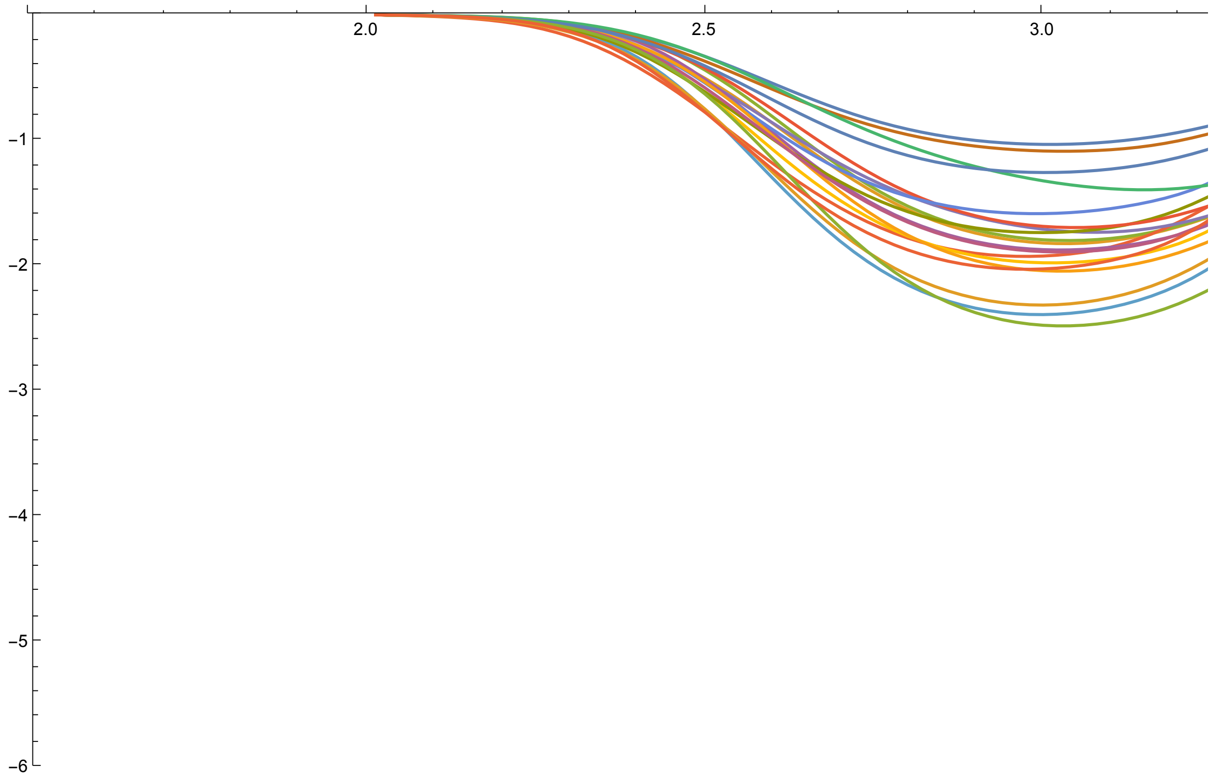
```
Dimensions[edgescan]
```

```
{16, 65}
```

Now generate 1000 waveforms with Landau fluctuations introduced.

```
waveforms = ConstantArray[0, {1000, 65}];
Do[
  (*clusters=RandomReal[LandauDistribution[μ,σ],16];*)
  clusters = RandomVariate[𝒟, 16];
  waveforms[[i]] = clusters.edgescan;
  , {i, 1, 1000}];
```

```
ListPlot[{Transpose[{time, waveforms[[1]]}],
  Transpose[{time, waveforms[[2]]}], Transpose[{time, waveforms[[3]]}],
  Transpose[{time, waveforms[[4]]}], Transpose[{time, waveforms[[5]]}],
  Transpose[{time, waveforms[[6]]}], Transpose[{time, waveforms[[7]]}],
  Transpose[{time, waveforms[[8]]}], Transpose[{time, waveforms[[9]]}],
  Transpose[{time, waveforms[[10]]}], Transpose[{time, waveforms[[11]]}],
  Transpose[{time, waveforms[[12]]}], Transpose[{time, waveforms[[13]]}],
  Transpose[{time, waveforms[[14]]}], Transpose[{time, waveforms[[15]]}],
  Transpose[{time, waveforms[[16]]}], Transpose[{time, waveforms[[17]]}],
  Transpose[{time, waveforms[[18]]}], Transpose[{time, waveforms[[19]]}]},
 Joined → True, PlotRange → {{1.5, 3.3}, {15 * (-.4), 0.}}]
```



```
MatrixPlot[waveforms, AspectRatio → 1]
```

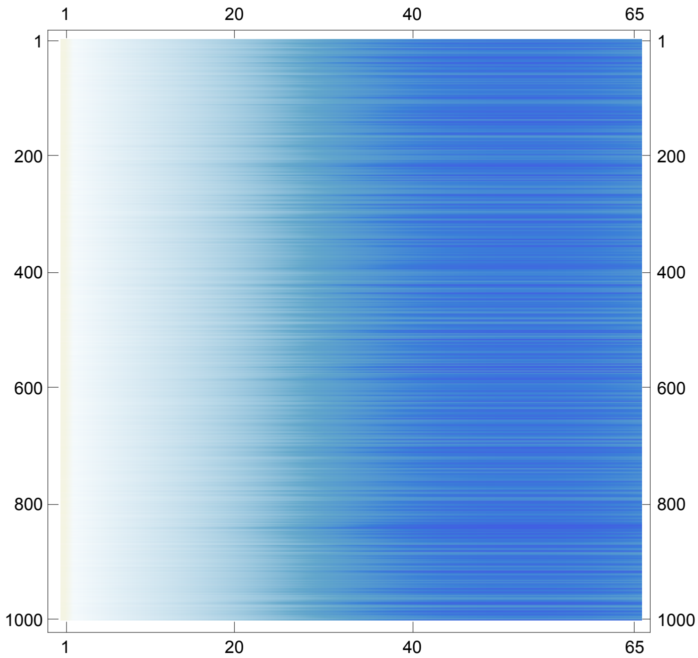


Figure 5

This (Figure 6) is the calculated MIP amplitude distribution with the Bichsel input- it is not a great match to the experimental plot (Figure 7) just below it.

```
peakvalue = Table[Max[-waveforms[[i]]] / 6.5, {i, 1, 1000}];
Histogram[peakvalue, {0, 0.6, 0.010}]
```

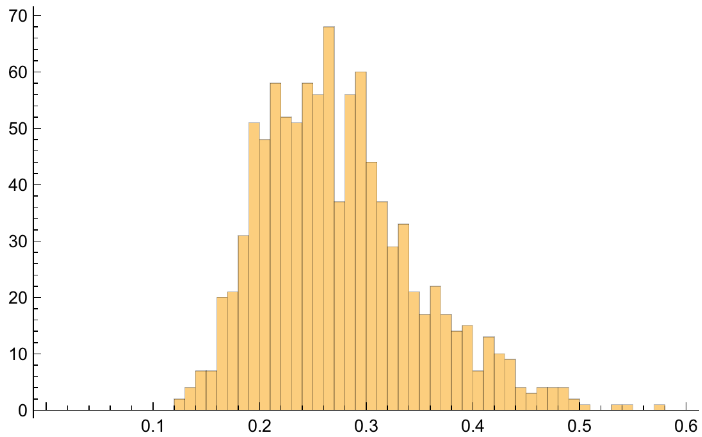


Figure 6

```
Export["simlandau2.dat", waveforms, "tsv"];
```

Out[786]=

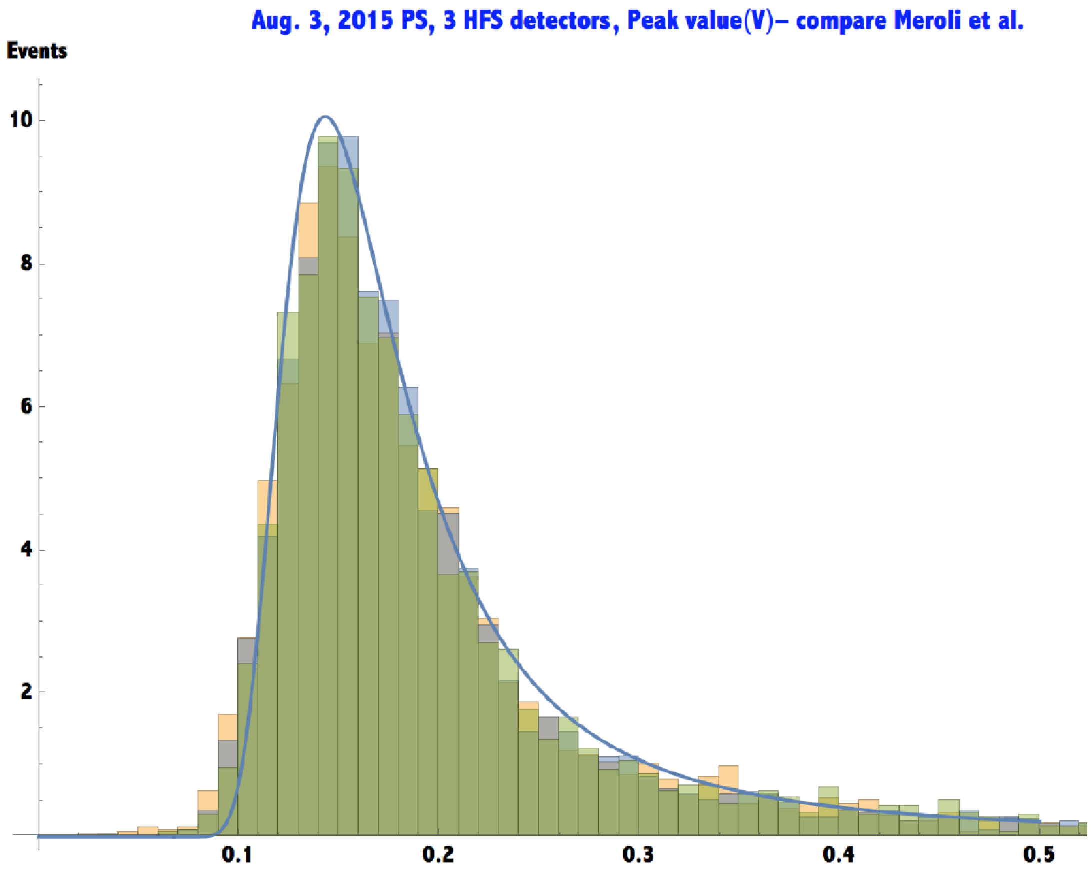
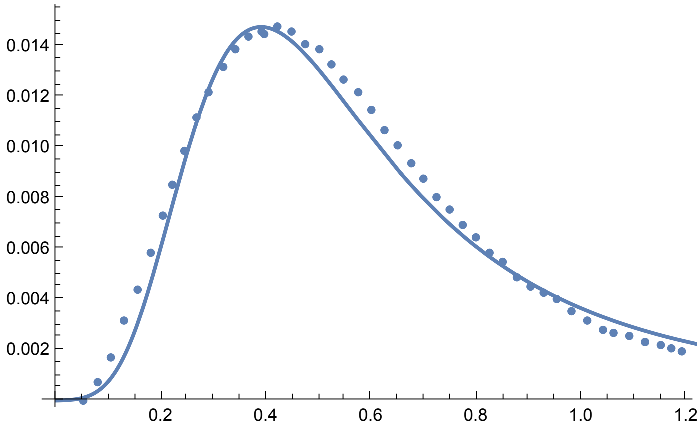


Figure 7

Below is an exercise in getting a best fit Landau form for the output of the Bichsel routine for 5 micron Silicon.

```
landau5micron = Import["xcelfromlu.csv", "csv"];
{x, y} = Transpose[landau5micron];
x = x / 100.;
```

```
σ = 0.2; μ = 0.68;
data = RandomReal[LandauDistribution[μ, σ], 10^6];
Show[
 ListPlot[Transpose[{x, y}], AxesOrigin → {0.0, 0}],
 Plot[PDF[LandauDistribution[μ, σ], x] / 96, {x, 0.0, 4},
  PlotRange → Full, PlotStyle → Thick, ImageSize → Large]]
```




```
Show[%514,
 AxesLabel → {HoldForm[Energy deposit - arb.scale], HoldForm[Frequency]},
 PlotLabel → HoldForm[Eyeball "Landau  fit" to Bichsel 5 micron],
 LabelStyle → {12, GrayLevel[0]}, ImageSize → Large, Background → LightGray]
```

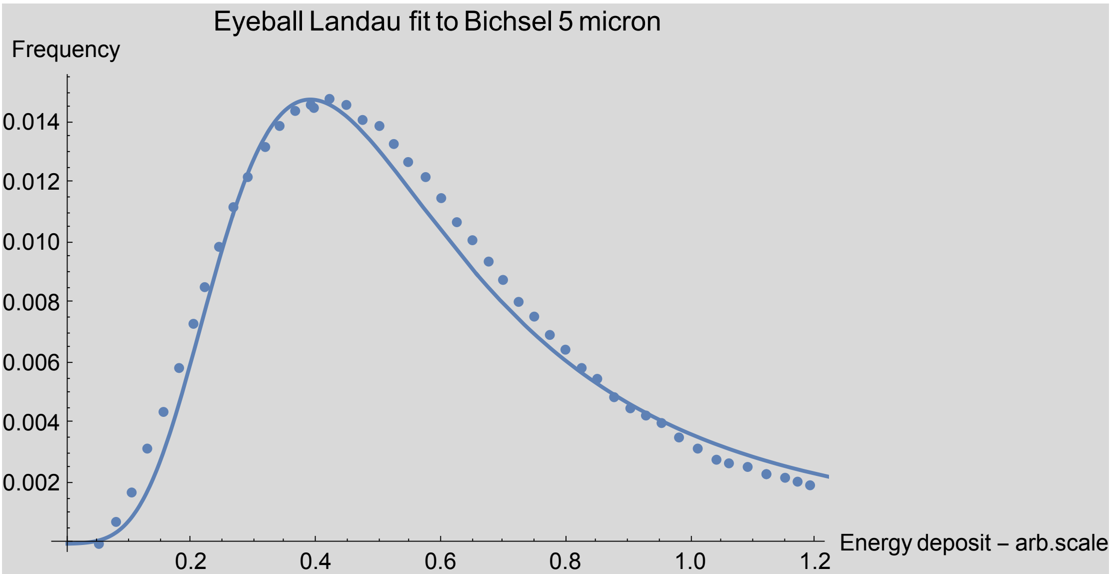


Figure 8

Below are results form Lu' s CF timing analysis on these waveforms :

Using Landau best fit to Bichsel output :

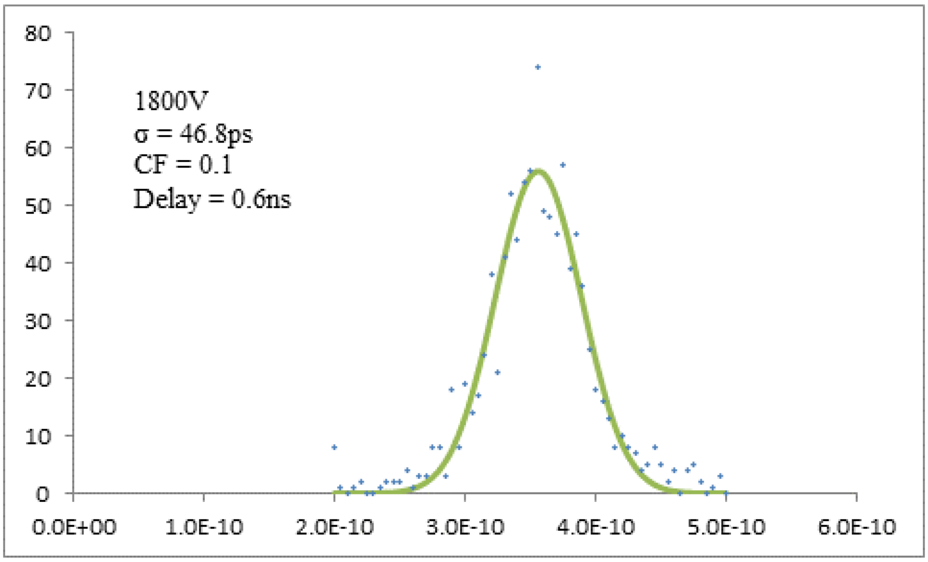


Figure 9

Using Bichsel exact output for pdf

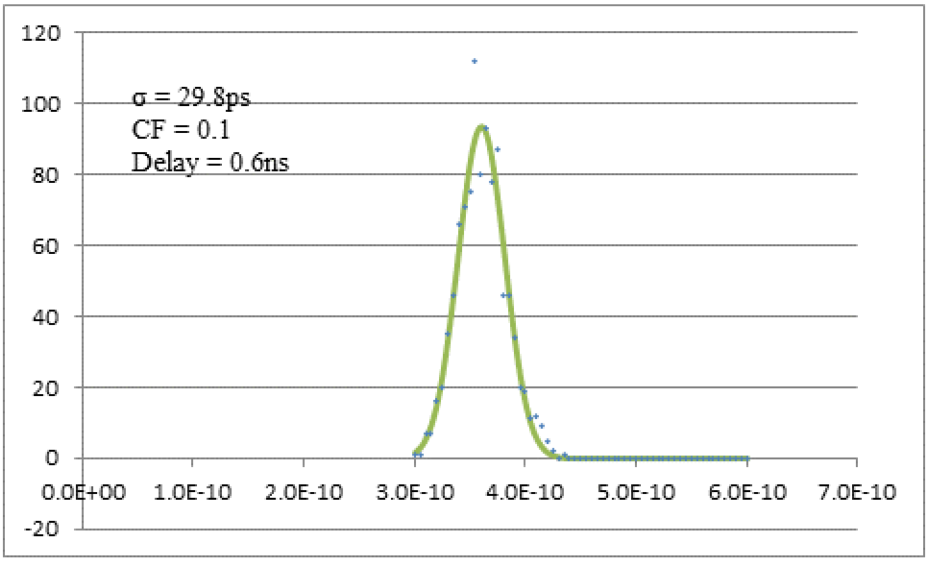


Figure 10

It is interesting that 2 such similar forms - shown in Fig.3 - give such dissimilar results - ie Figures 9 and Fig.10!