

Final results

Beamcats

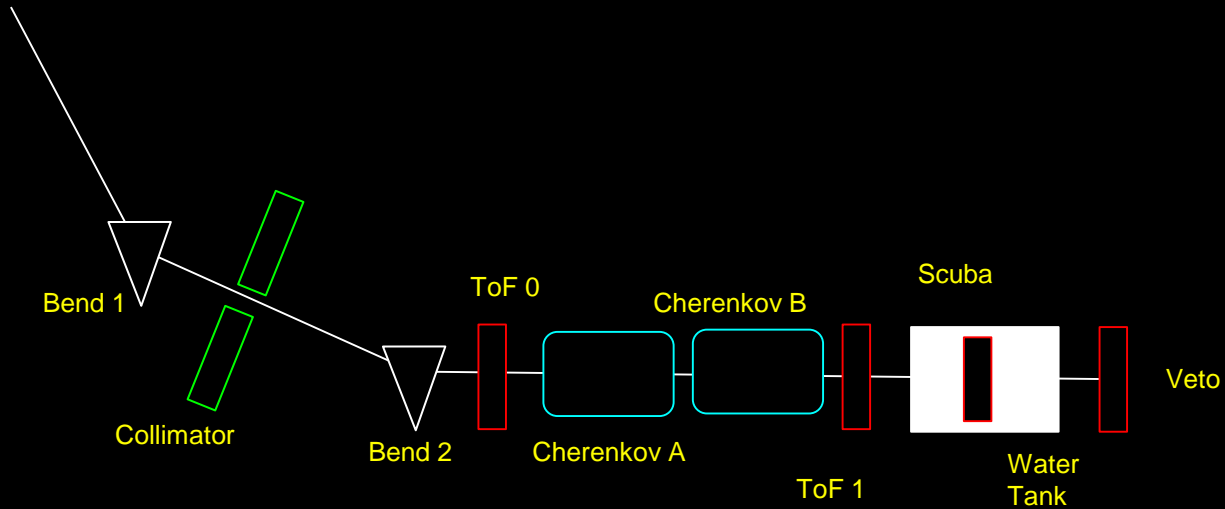
Charvie, Sana, Aarushi, Yash, Ashish, Saejoon



Our Experience at CERN



New Experimental Setup



Our Process



Coding

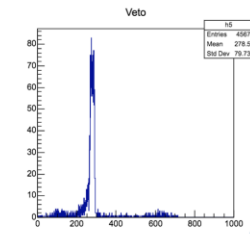
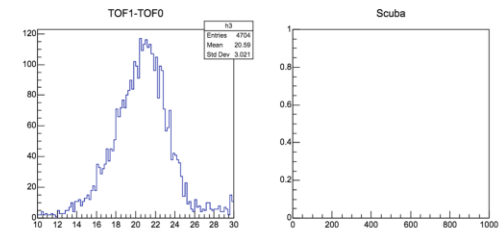
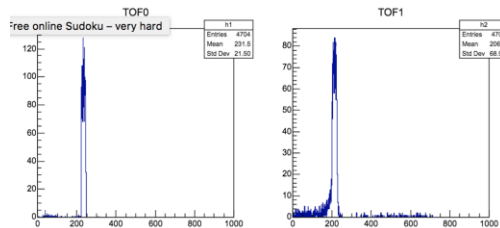
```
from ROOT import TFile, TH1F, TCanvas, enableJSVis #import stuff
c1 = TCanvas("c1","Practice Graph",1200,400) #Makes a blank canvas
c1.Divide(3,2) #Divides the canvas into 4 columns, 4 rows

h1 = TH1F("h1","TOF0",2000,0,1000) #Makes a blank histogram: TH1F("name", "Title", "bins", "min", "max")
h2 = TH1F("h2","TOF1",2000,0,1000) #Makes a blank histogram: TH1F("name", "Title", "bins", "min", "max")
h3 = TH1F("h3","TOF1-TOF0",100,10,30) #Makes a blank histogram: TH1F("name", "Title", "bins", "min", "max")
h4 = TH1F("h4","Scuba",2000,0,1000)
h5 = TH1F("h5","Veto",2000,0,1000)

b183c0= oetmm.GetBranch("NTDC0_ch0")
b183c1= oetmm.GetBranch("NTDC0_ch1")
b183c2= oetmm.GetBranch("NTDC0_ch3")
b183c3= oetmm.GetBranch("NTDC0_ch4")
b1830= oetmm.GetBranch("TDC0_ch0")
b1831= oetmm.GetBranch("TDC0_ch1")
b1832= oetmm.GetBranch("TDC0_ch2")
b1833= oetmm.GetBranch("TDC0_ch3")

for i in range(oetmm.GetEntries()):
    b183c0.GetEntry(i)
    p0 = oetmm.NTDC0_ch0
    b183c1.GetEntry(i)
    p1 = oetmm.NTDC0_ch1
    b183c2.GetEntry(i)
    p2 = oetmm.NTDC0_ch2
    b183c3.GetEntry(i)
    p3 = oetmm.NTDC0_ch3
    if 0< p0 <2 and 0< p1 < 2:
        b1830.GetEntry(i)
        val0 = oetmm.TDC0_ch0[0] * 0.025
        h1.Fill(val0)
        b1831.GetEntry(i)
        val1 = oetmm.TDC0_ch1[0] * 0.025
        h2.Fill(val1)
        valmin = val0 - val1
        h3.Fill(valmin)
    if 0< p2 <2:
        b1832.GetEntry(i)
        val2 = oetmm.TDC0_ch2[0] * 0.025
        h4.Fill(val2)
    if 0< p3 <2:
        b1833.GetEntry(i)
        val3 = oetmm.TDC0_ch3[0] * 0.025
        h5.Fill(val3)

ROOT.enableJSVis() #JAVA SCRIPT MODE is interactive
c1.Draw() #draws canvas
c1.cd(1) #moves to space number 1
h1.Draw() #draws histogram
c1.cd(2)
h2.Draw()
c1.cd(3)
h3.Draw()
c1.cd(4)
h4.Draw()
c1.cd(5)
h5.Draw()
```





Coding

```
def draw2D(filename, savenamed,option):
    import ROOT
    from ROOT import TFile, TH2F, TCanvas, enableJSVis #import stuff

    c2 = TCanvas("c2","",600,400) #Makes a blank canvas

    fild = ROOT.TFile("/eos/user/d/daquser/2018/Bragg/%s" %(filename))
    trd = fild.Get("RAWdata")
    h6 = TH2F("h6","Charge VS. Time Difference",600,-300,300,4000,0,4000)

    bc0= trd.GetBranch("NTDC0_ch0")
    bc1= trd.GetBranch("NTDC0_ch1")
    bdc2= trd.GetBranch("NTDC0_ch2")
    b0= trd.GetBranch("TDC0_ch0")
    b1= trd.GetBranch("TDC0_ch1")
    bd2= trd.GetBranch("QDC0_ch2")

    for i in range(trd.GetEntries()):
        bc0.GetEntry(i)
        p0 = trd.NTDC0_ch0
        bc1.GetEntry(i)
        p1 = trd.NTDC0_ch1
        bdc2.GetEntry(i)
        p2 = trd.NTDC0_ch2

        if 0< p0 <2 and 0< p1 < 2 and p2 <2:
            b0.GetEntry(i)
            vald0 = trd.TDC0_ch0[0] * 0.025
            b1.GetEntry(i)
            vald1 = trd.TDC0_ch1[0] * 0.025
            valdmin = vald1 - vald0
            bd2.GetEntry(i)
            vald2 = trd.QDC0_ch2
            h6.Fill(valdmin, vald2)

    ROOT.enableJSVis() #JAVA SCRIPT MODE is interactiveabl
    h6.Draw(option) #draws histogram
    c2.Draw()

    c2.SaveAs(savenamed + ".pdf")
    c2.SaveAs(savenamed + "graph")

    return savenamedgraph
```

```
def drawcharge(filename, savenameq, rangemin, rangemax):
    import ROOT
    from ROOT import TFile, TH1F, TCanvas, enableJSVis #import stuff

    c3 = TCanvas("c3","",600,400) #Makes a blank canvas

    filq = ROOT.TFile("/eos/user/d/daquser/2018/Bragg/%s" %(filename))
    trq = filq.Get("RAWdata")
    h7 = TH1F("h7","Charge for Protons",200,0,4500) #Makes a blank his

    bc0= trq.GetBranch("NTDC0_ch0")
    bc1= trq.GetBranch("NTDC0_ch1")
    bq2= trq.GetBranch("NTDC0_ch2")
    bc3= trq.GetBranch("NTDC0_ch3")
    b0= trq.GetBranch("TDC0_ch0")
    b1= trq.GetBranch("TDC0_ch1")
    bq2= trq.GetBranch("QDC0_ch2")

    for i in range(trq.GetEntries()):
        bc0.GetEntry(i)
        p0 = trq.NTDC0_ch0
        bc1.GetEntry(i)
        p1 = trq.NTDC0_ch1
        bq2.GetEntry(i)
        p2 = trq.NTDC0_ch2
        bc3.GetEntry(i)
        p3 = trq.NTDC0_ch3

        if 0< p0 <2 and 0< p1 < 2 and 0< p2 <2 and p3 == 0:
            b0.GetEntry(i)
            valq0 = trq.TDC0_ch0[0] * 0.025
            b1.GetEntry(i)
            valq1 = trq.TDC0_ch1[0] * 0.025
            valqmin = valq1 - valq0
            bq2.GetEntry(i)

            if rangemin < valqmin < rangemax:
                valq2 = trq.QDC0_ch2
                h7.Fill(valq2)

    ROOT.enableJSVis() #JAVA SCRIPT MODE is interactiveabl
    h7.Draw() #draws histogram
    c3.Draw()

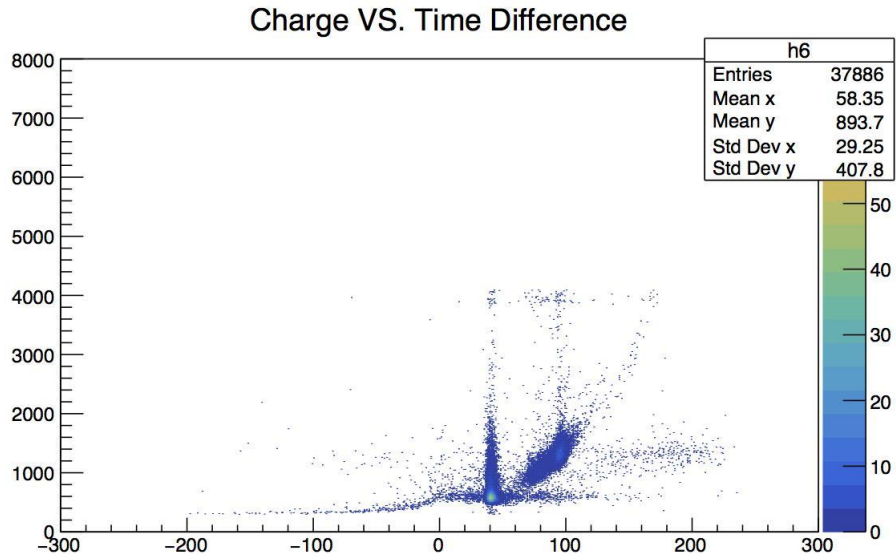
    c3.SaveAs(savenameq + ".pdf")
    c3.SaveAs(savenameq + "graph")

    return savenameqgraph
```



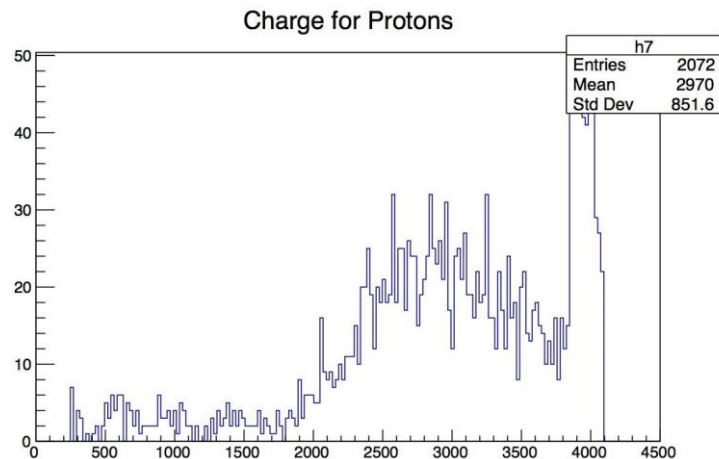
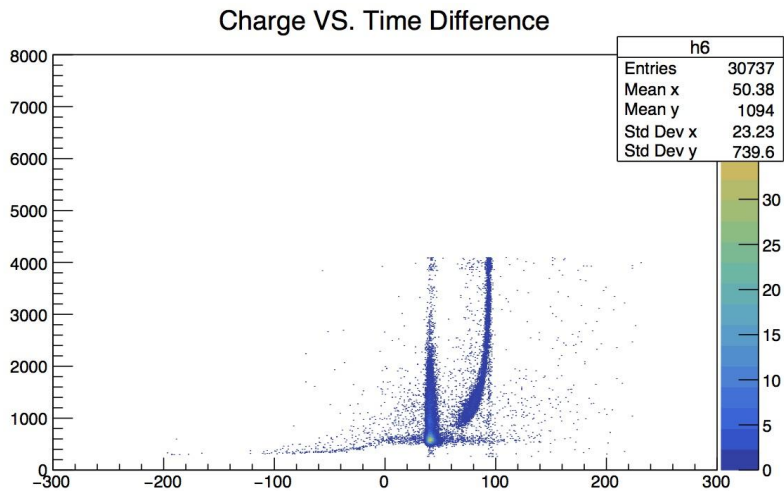
Challenge #1: Calibration

- Collected Data from Monday to Tuesday
 - Realised there were high levels of noise

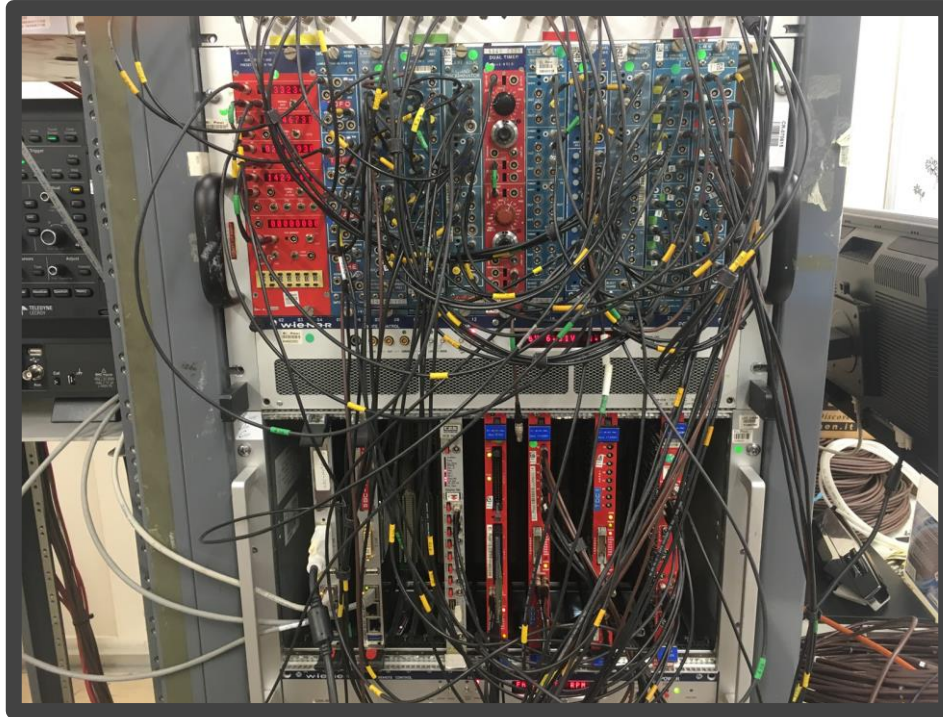


Challenge #2: Saturation

- Collected Data from Wednesday to Thursday
 - Realised that there was saturation of the data

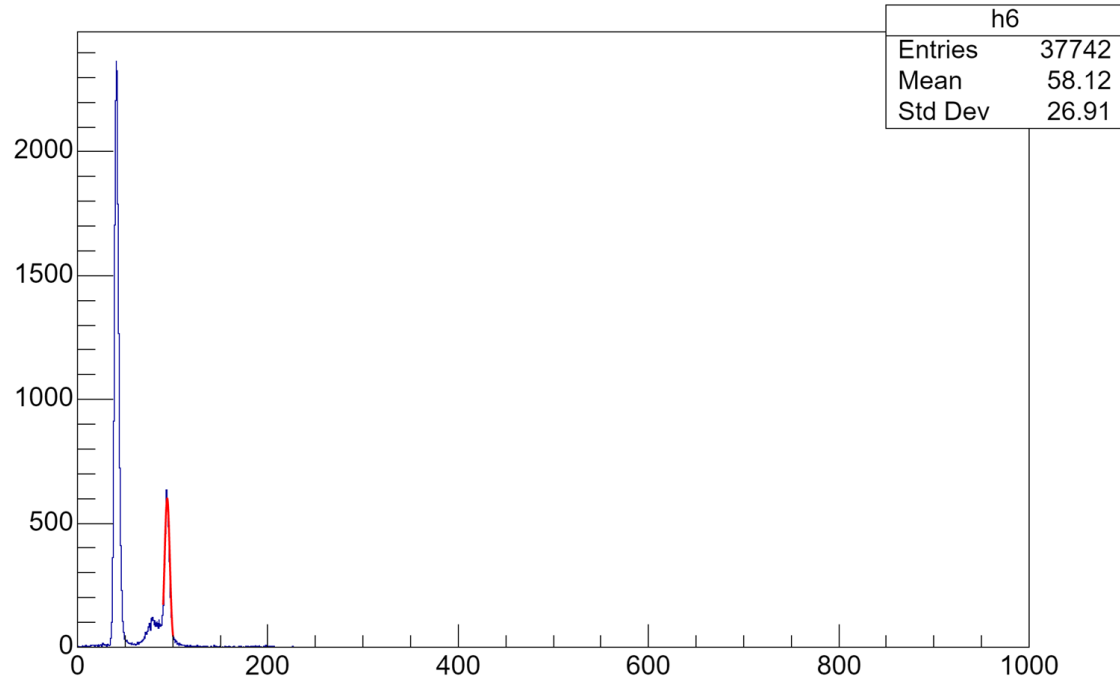


Challenge #3: Out of Sync Sensors



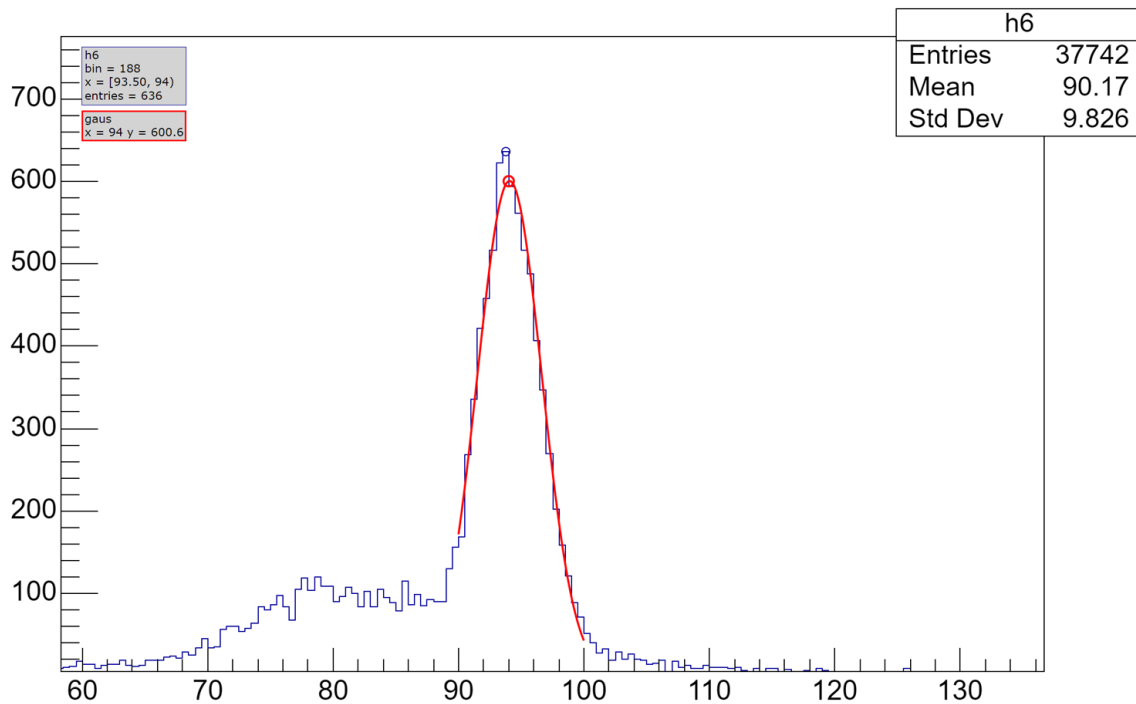


ToF Plots





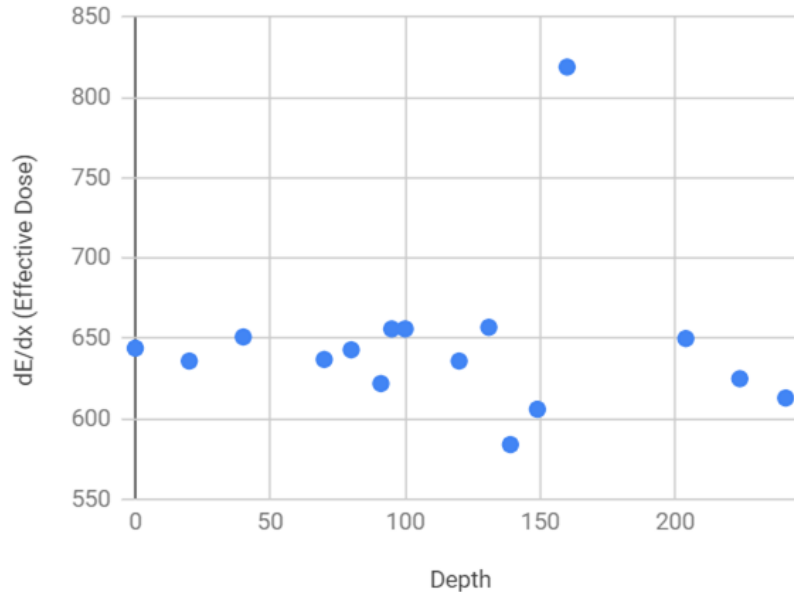
ToF Plots





Results, 600 MeV

dE/dx vs. Depth



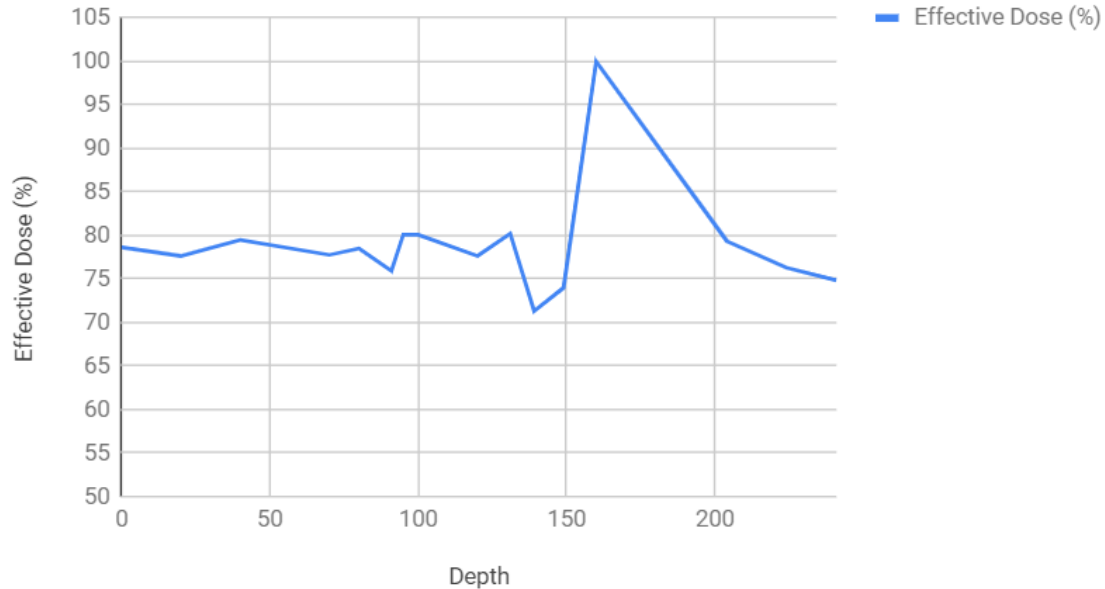
- Unable to set distances at a precision of more than 5 mm, resolution effected
- Random errors: background noise, scattering.

Depth	Energy
0	644
20	636
40	651
70	637
80	643
91	622
95	656
100	656
120	636
131	657
139	584
149	606
160	819
204	650
224	625
241	613



Results, 600 MeV Percentage

Effective Dose (%) vs Depth





Looking Ahead