Status of ray trace background

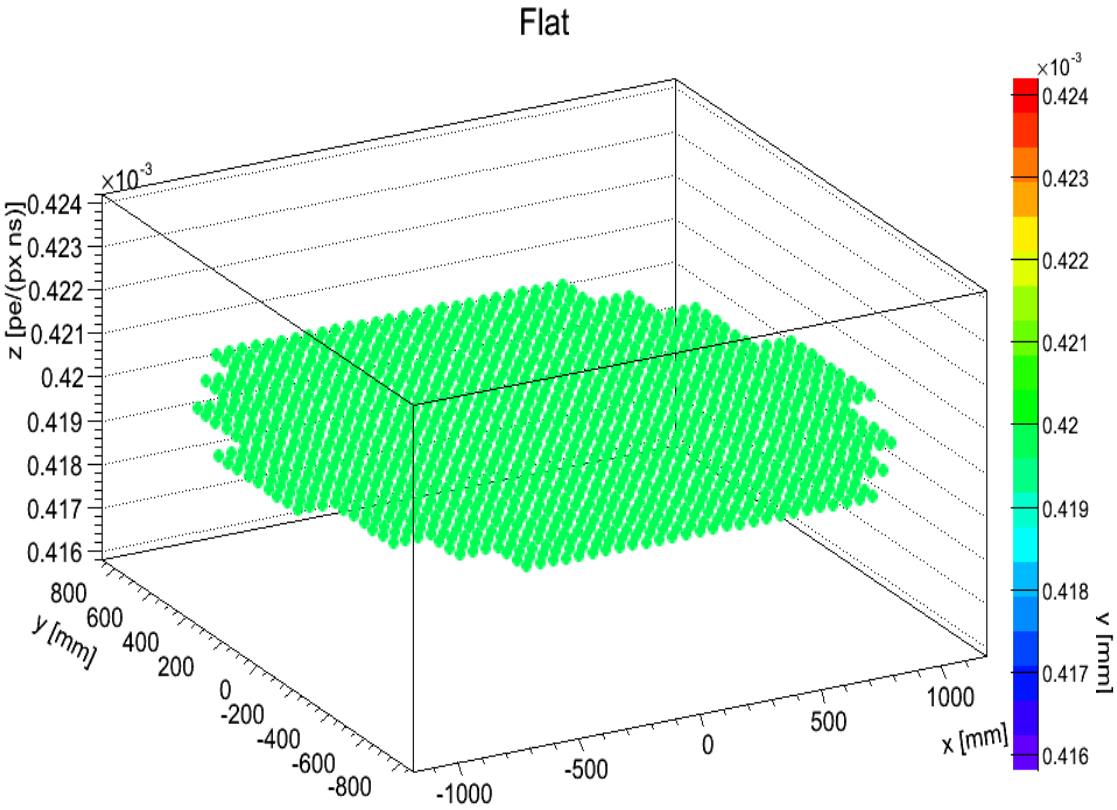
P. Bobik, L. Gombar, K. Kudela,
B. Pastircak, M. Putis

Current status of code

- Class for background calculations: EusoElectronics
- Two main options for Night Glow background:
  - byRate
  - byRadiance (not used in this study)
- byRate – options for shape:
  - Flat – 0.42 pe/(px μs) for each elementary cell
  - CosTheta – cos(θ) dependence of rate, (0.42 pe/(px μs) refers to theta 0)
  - New switch: RayTraceBG – load value from ray trace simulation for each EC from data file. (config/Electronics/EusoElectronics/NightGlow_RayTraceBG.dat)

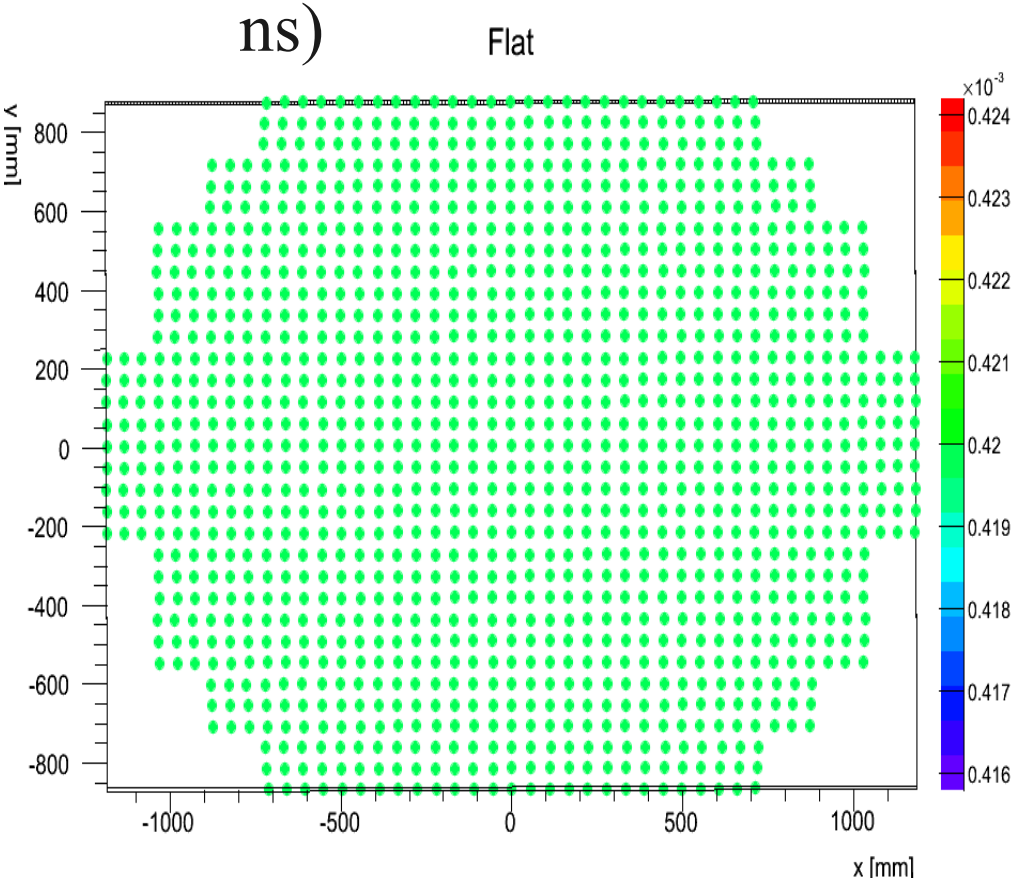
- Configuration for this study:
  - temporal_complutensian_MAR2012_PPP2010_ammend_v1.cfg

Flat

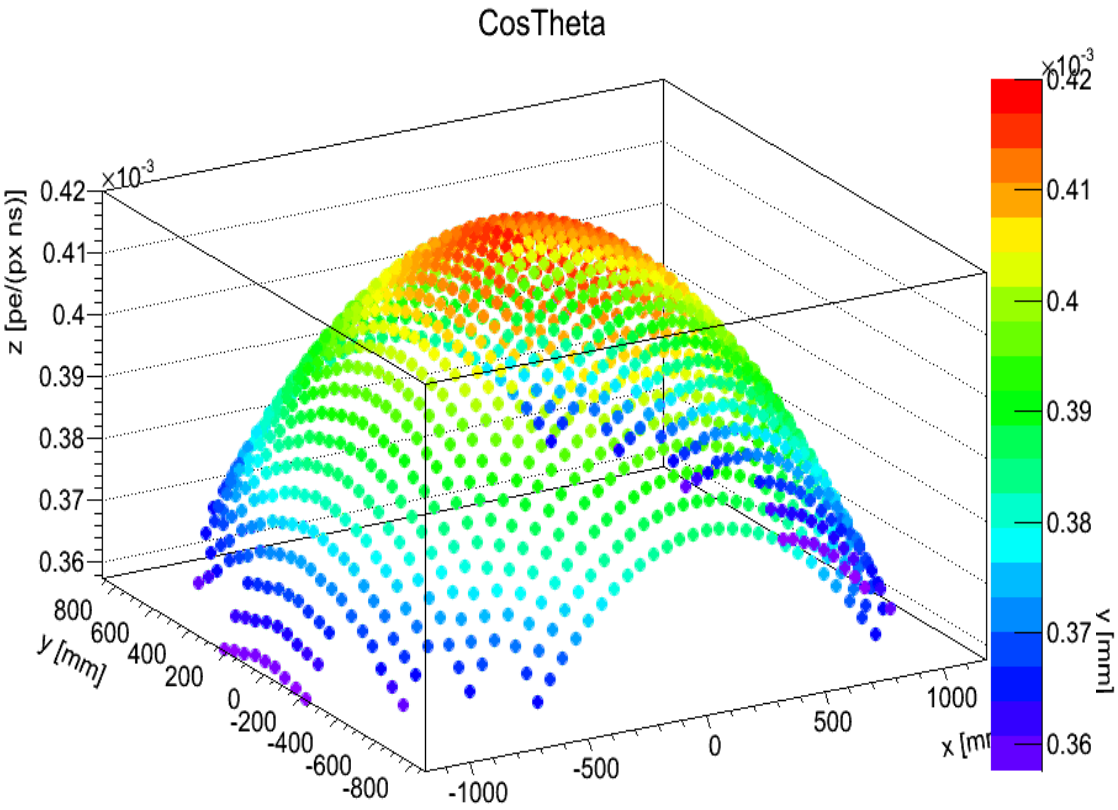


- 0.42 pe/(px μs) for each elementary cell

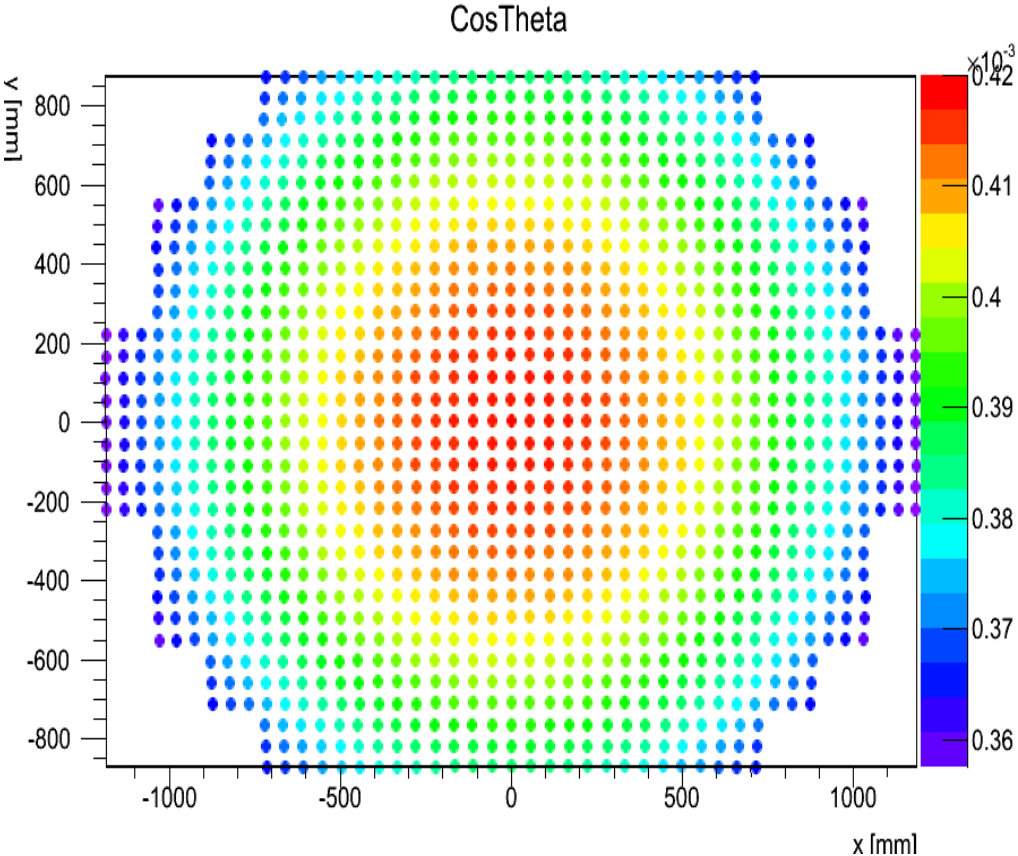
- x,y axis - position of center of EC in mm
- z axis (also color palette) show values of background in pe/(px ns)



CosTheta

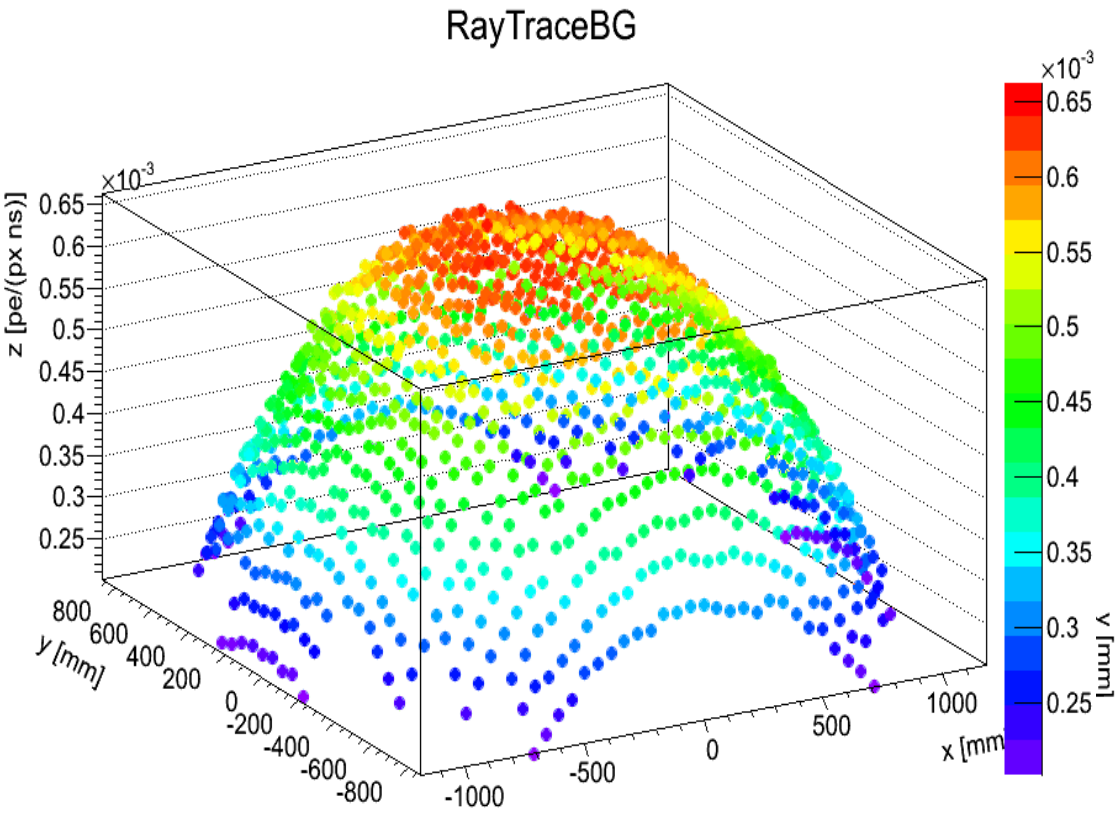


- CosTheta dependence of rate
- 0.42 pe/(px μs) corresponding to theta 0



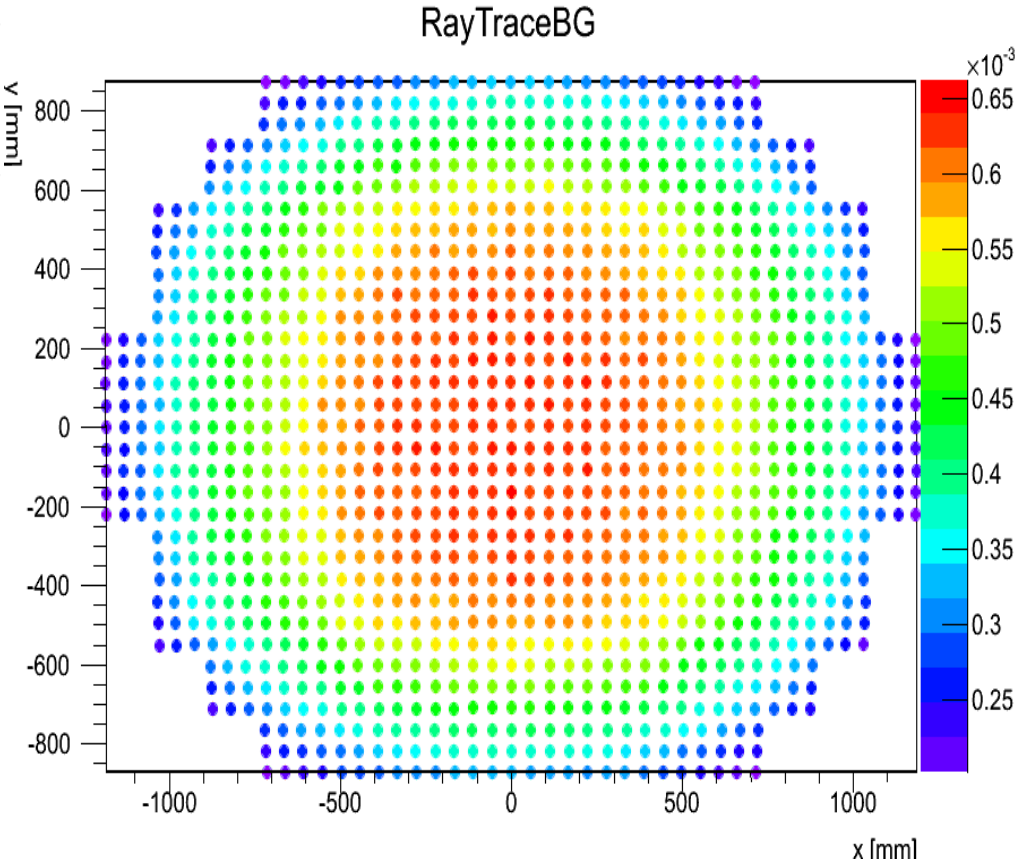
- Range of values ~0.36 - 0.42 pe/(px μs)

RayTraceBG

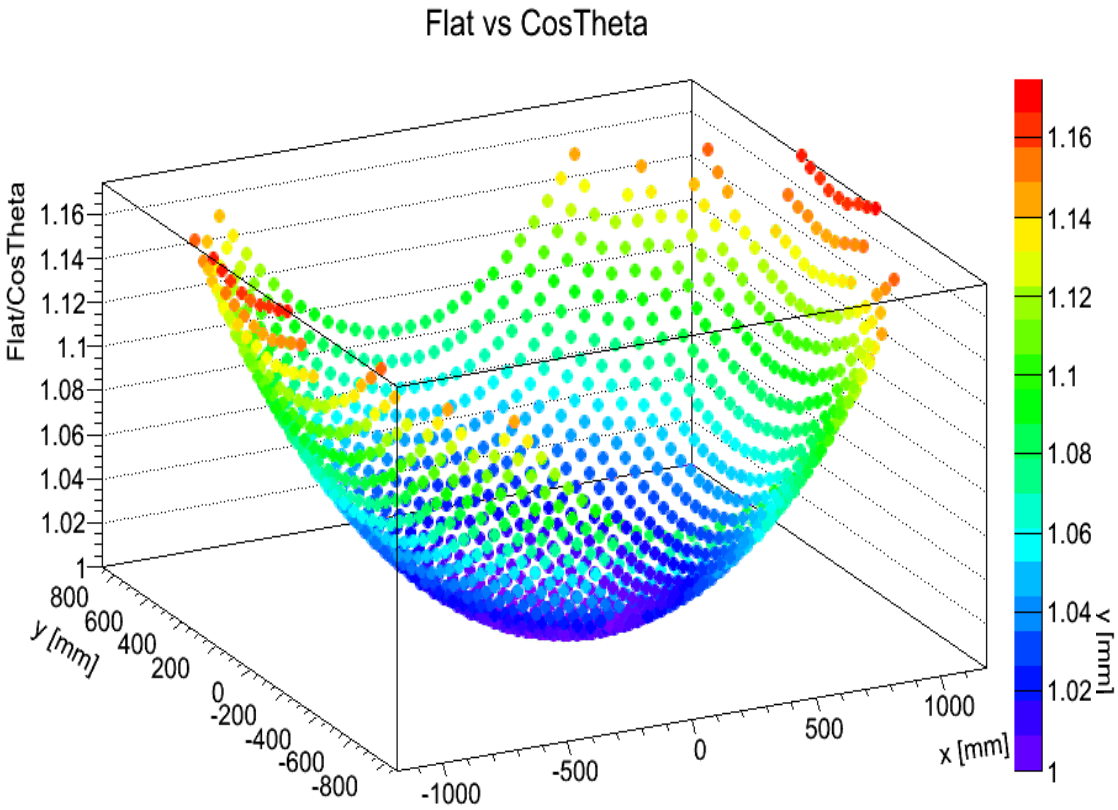


- Range of values ~ 0.2 - 0.66 pe/(px μs)
- Average value: ~ 0.47 pe/(px μs)

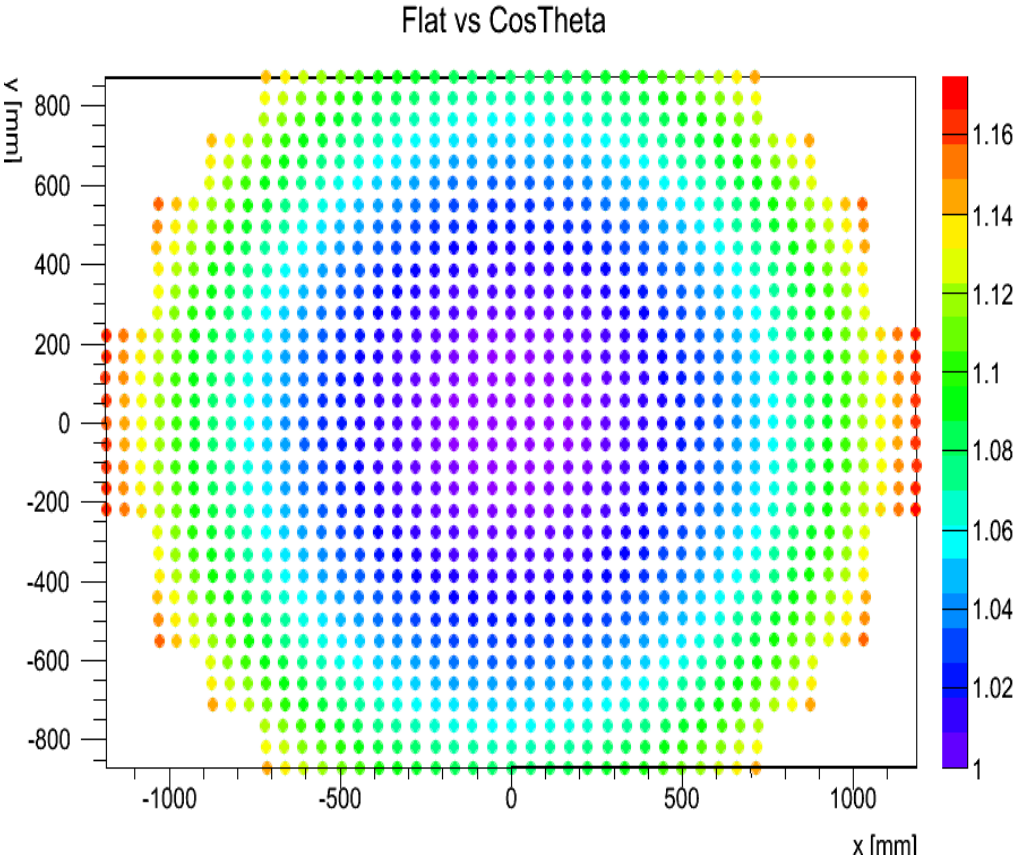
- For more details about ray trace simulations see presentation: P. Bobik JEM-EUSO simulation meeting, Madrid, *26.-30. march 2012*



Flat vs CosTheta

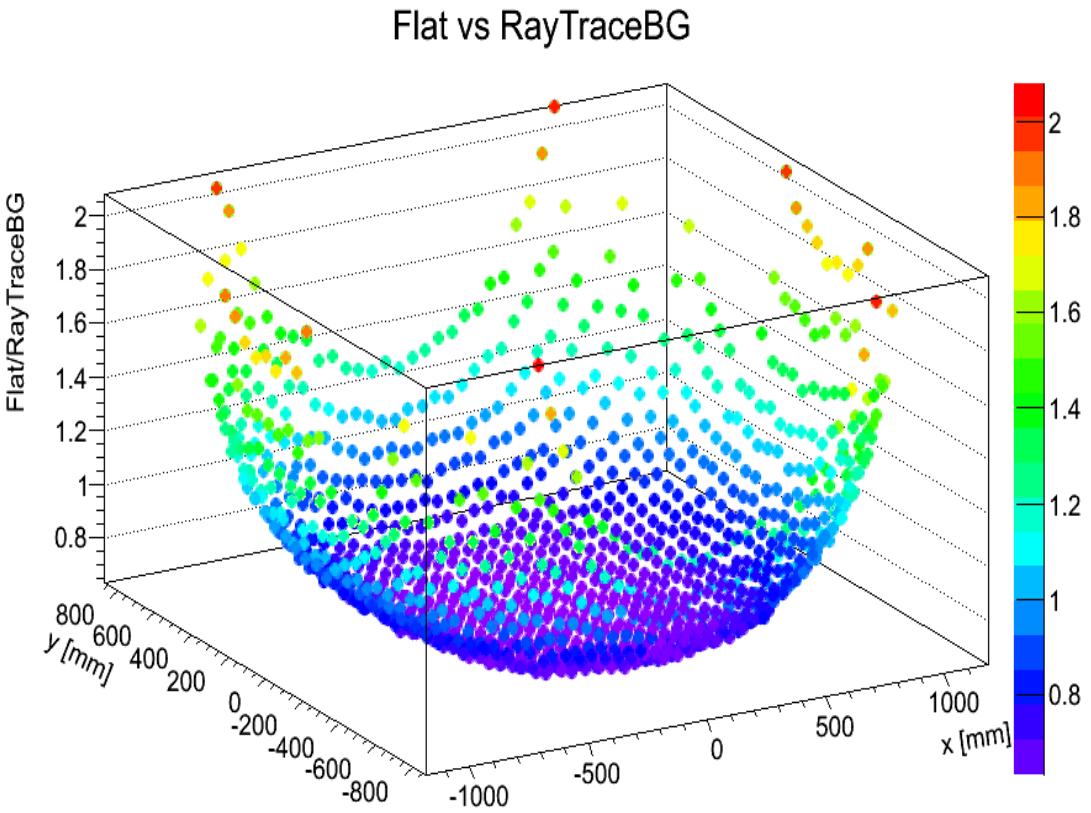


- z axis (also color palette) Flat/CosTheta

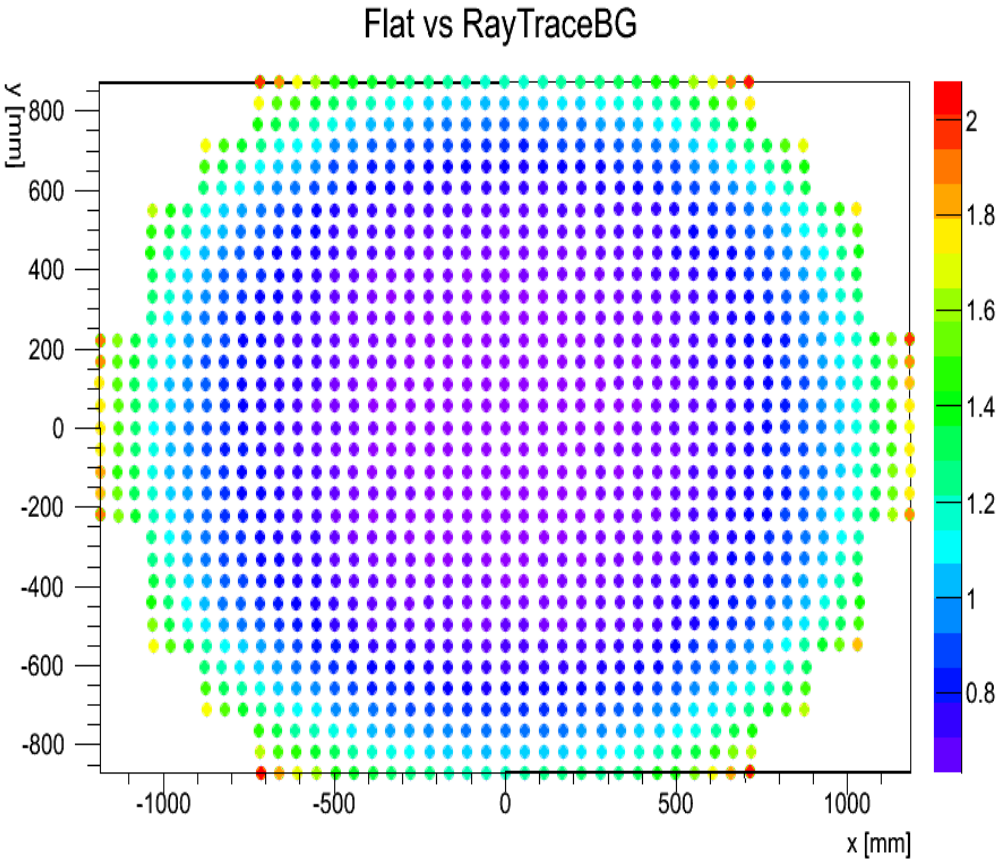


- Ratio:
  - In center ~ 1 for $\theta = 0$
  - On the edge ~ 1.17

Flat vs RayTraceBG

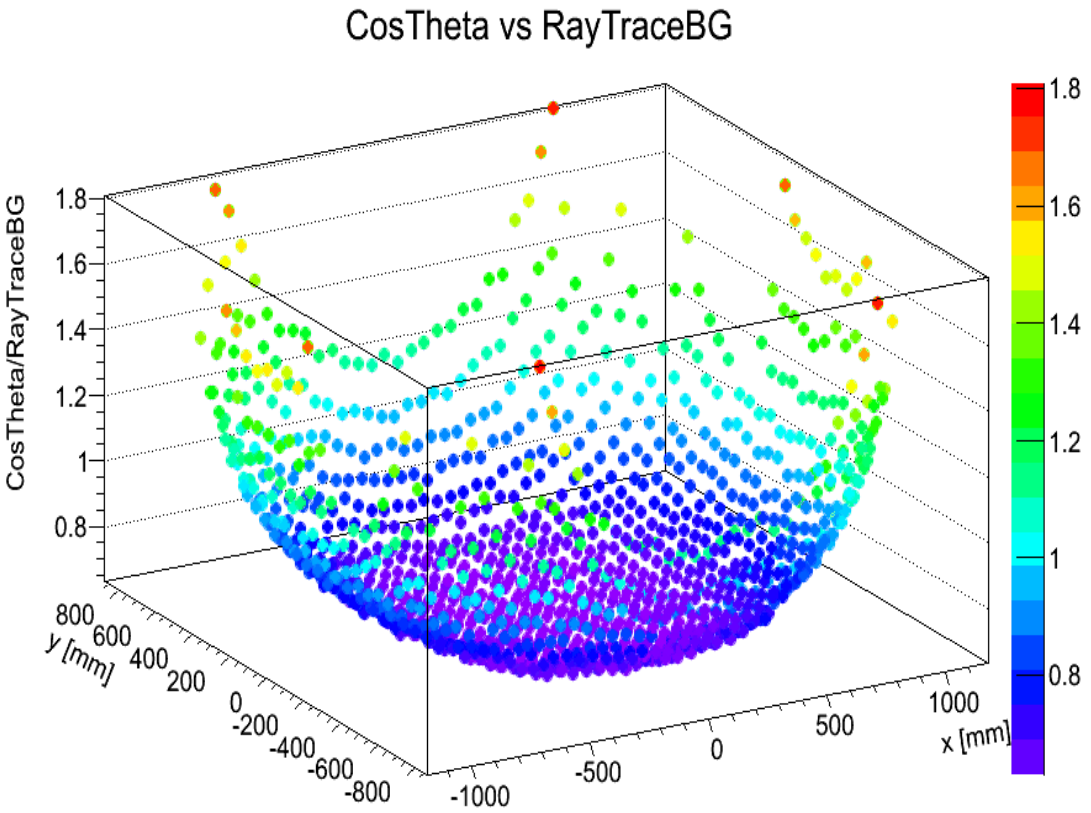


- z axis (also color palette) Flat/RayTraceBG

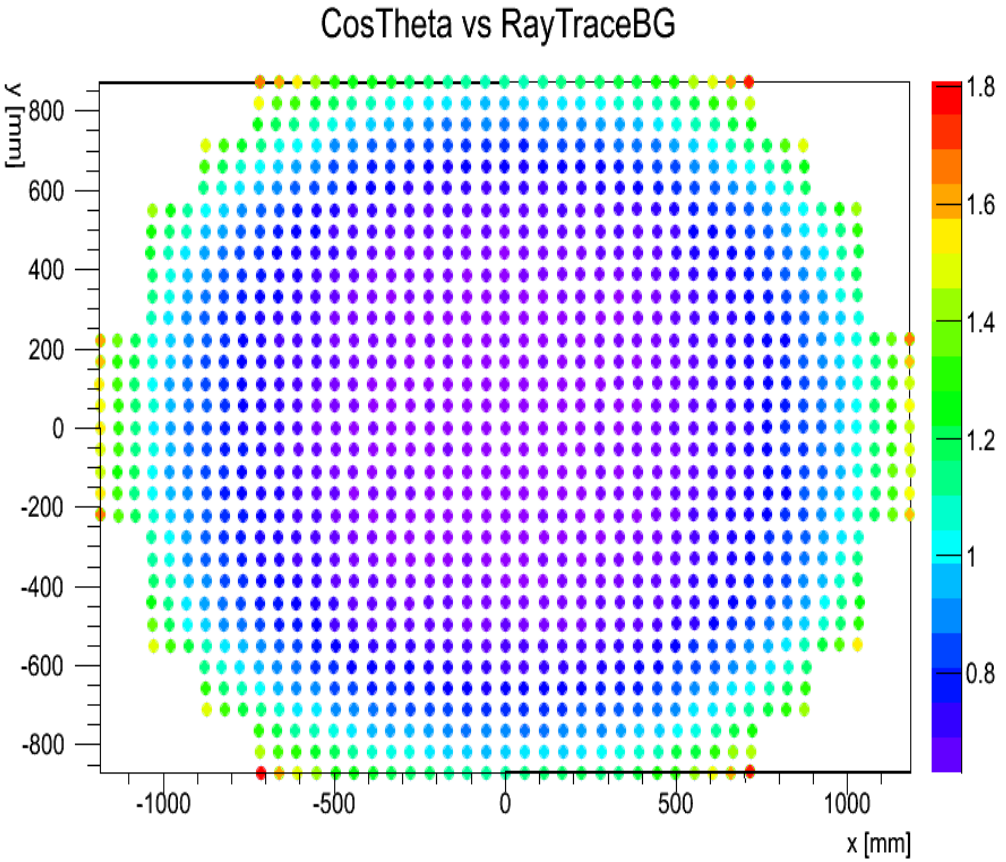


- Ratio:
  - In center ~ 0.65
  - On the edge ~ 2.05 for few EC

CosTheta vs RayTraceBG



- z axis (also color palette) CosTheta/RayTraceBG



- Ratio:
  - In center ~ 0.65
  - On the edge ~ 1.8 for few EC

ERayTraceReader.cc

```
#include <iostream>
#include "ERayTraceReader.hh"
#include <fstream>

using namespace std;
ClassImp(ERayTraceReader)

  ERayTraceReader::ERayTraceReader() //constructor
  {
                x_mm = new Double_t [1233];
                y_mm = new Double_t [1233];
                value_xy_mm = new Double_t [1233];

        for (Int_t j=0;j<1233;j++){
                x_mm[j]=0;
                y_mm[j]=0;
                value_xy_mm[j]=0;  }

  }

  ERayTraceReader::~ERayTraceReader() //destructor
  {
        delete [] x_mm;
        delete [] y_mm;
        delete []  value_xy_mm;
  }

  void ERayTraceReader::LoadFromFile_mm(const Char_t *lnfe)
  {
    Int_t i=0;
    Double_t x,y,value=0;

    ifstream myfile1;
    myfile1.open(lnfe); // open file to read data

          while(!myfile1.eof()) {

                myfile1 >>x;
                myfile1 >>y;
                myfile1 >>value;
                x_mm[i]=x;
                y_mm[i]=y;
                value_xy_mm[i]=value;
                i=i+1;
                if (i==1233) break;
                  }
  myfile1.close();
  }
```

ERayTraceReader.hh

```
#ifndef _ERAYTRACEREADER_H_
#define _ERAYTRACEREADER_H_

#include "TObject.h"

  class ERayTraceReader
  {

  public:

         ERayTraceReader();
         virtual  ~ERayTraceReader();
         void LoadFromFile_mm(const Char_t * lnfe);
         virtual Double_t GetX(Int_t j) { return fx=x_mm[j]; }
         virtual Double_t GetY(Int_t j) { return fy=y_mm[j]; }
         virtual Double_t GetValueXY(Int_t j) { return fv=value_xy_mm[j]; }

  private:
          Double_t *x_mm;
          Double_t *y_mm;
          Double_t *value_xy_mm;
          Double_t fx;
          Double_t fy;
          Double_t fv;

    ClassDef(ERayTraceReader,1)
  };


#endif
```

EusoElectronics::Bulid

```
    if (fNightGlowShape =="RayTraceBG"){

         fGtu=0;
         string path_RayTraceBG = Conf()->GetCfgDir()+'/'+
             ClassType()+'/'+
             ClassName()+'/'+"NightGlow_"+fNightGlowShape+".dat";
             const Char_t *path_RayTraceBG_Ch;
             path_RayTraceBG_Ch=path_RayTraceBG.c_str();

         fNightGlowrayTraceBG = new ERayTraceReader();
         fNightGlowrayTraceBG->LoadFromFile_mm(path_RayTraceBG_Ch);
         fGtu= Config::Get()->GetCF("Electronics","MacroCell")->GetNum("MacroCell.fGtuTimeLength");

                                    }
```

EusoElectronics::NightGlowRate

```
//_____________________________________________________________________________
Double_t EusoElectronics::NightGlowRate( const TVector3& pos, const TVector3& norm,
         Double_t pxsize, Double_t pde ) const {
    //
    // Nightglow rate as function of the location on the FS
    // (number of hits per microseconds)

    Double_t ngr(0),x(0),y(0),xx(0),yy(0);

    if ( fNightGlow == "byRate" ) {
        if ( fNightGlowShape == "Flat" )
            ngr = fNightGlowRateOnAxis;
        else if ( fNightGlowShape == "CosTheta" ) {
            Double_t fsPosZ = 2200*mm;

            Double_t tg2th = pos.Perp2();//(pos.x()*pos.x() + pos.y()*pos.y());
            tg2th /= ( (fsPosZ+pos.z()) * (fsPosZ+pos.z()));
            Double_t cth = 1. / TMath::Sqrt((1. + tg2th));
            ngr = fNightGlowRateOnAxis*cth;

        }else if (fNightGlowShape == "RayTraceBG")  {
                x=pos.x();
                y=pos.y();

        for (Int_t i=0; i<1233;i++){
                xx=fNightGlowrayTraceBG->GetX(i);
                yy=fNightGlowrayTraceBG->GetY(i);

            if ( (x<xx+0.1) && (x>xx-0.1) && (y < yy+0.1) && (y>yy-0.1) ) {
                ngr=(fNightGlowrayTraceBG->GetValueXY(i))*(pxsize*pxsize/fGtu); //conversion from pe/(mm2 GTU) on pe/(px ns), GTU is in [ns]
                                                                                }
                        }
        }
         else
            FatalError("Unknown night glow shape:"+fNightGlowShape);
    } else if ( fNightGlow == "byRadiance" ) {
        Double_t r = pos.Perp();

        if ( r > fNightGlowDist->GetXmax() ) {
            ngr = 0;
        } else {
            ngr = fNightGlowDist->GetValue( r )*fNightGlowRadiance;

            // apply quantum efficiency and pmt orientation
            ngr *= pxsize*pxsize*Abs(norm.CosTheta());
            ngr *= pde;
            ngr *= fDetectorScaleFactor*fDetectorScaleFactor;

        }

    }
    return ngr;
```

Conclusion

- Ray trace background have more sharp shape than CosTheta background.

- Flat/CosTheta: In center ~ 1, On the edge ~ 1.17
- Flat/RayTraceBG: In center ~ 0.65, On the edge ~ 2.05
- CosTheta/RayTraceBG: In center ~ 0.65, On the edge ~ 1.8

- Need to be recalculated?
  - Range of wavelength for RT BG: 300 – 400 nm
  - Range of wavelength in config. file: 250 – 485 nm
  - What is the final wavelength range?