```
void tviewer() {
  //  TBrowser *tb = new TBrowser();
  gSystem->Load("libTreeViewer.so");
  new TTreeViewer();
  return;

}
void tbrowser() {
  TBrowser *tb = new TBrowser();
  //  gSystem->Load("libTreeViewer.so");
  //  new TTreeViewer();
  return;
}

void rtheta_e(float ec) {

  float theERange = ec;
  char chfich[100];
  sprintf(chfich,"HistElectron_acc_%3.1f_MeV.root",ec);
  printf("chfich %s\n",chfich);
  TFile *f = new TFile(chfich);
  f->ls();

  char chist[60];
  char chist2[60];
  char ctext[60];
  float ce[15] = {2.0, 3.0, 5.0, 10.0, 15.0, 25.0, 35.0, 45.0, 55.0, 75.0, 100.0
,
                  125.0, 150.0, 200.0, 1.5};

  TH2D *hrtheta2d[3];
  TH1D *hproj;
  TH1D *hrtheta[3];
  TLatex *trtheta;

  char chistg[10];
  char chista[10];
  char chistasv[10];

  if(theERange == 2.0){
    sprintf(chistg,"h1000");
    sprintf(chista,"h1100");
    sprintf(chistasv,"h1200");
    sprintf(ctext,"T_{e} = %.1f MeV",ce[0]);
  }else if(theERange == 3){
    sprintf(chistg,"h1001");
    sprintf(chista,"h1101");
    sprintf(chistasv,"h1201");
    sprintf(ctext,"T_{e} = %.1f MeV",ce[1]);
  }else if(theERange == 5){
    sprintf(chistg,"h1002");
    sprintf(chista,"h1102");
    sprintf(chistasv,"h1202");
    sprintf(ctext,"T_{e} = %.1f MeV",ce[2]);
  }else if(theERange == 10){
    sprintf(chistg,"h1003");
    sprintf(chista,"h1103");
    sprintf(chistasv,"h1203");
    sprintf(ctext,"T_{e} = %.1f MeV",ce[3]);
  }else if(theERange == 15){
    sprintf(chistg,"h1004");
    sprintf(chista,"h1104");
    sprintf(chistasv,"h1204");
    sprintf(ctext,"T_{e} = %.1f MeV",ce[4]);
  }else if(theERange == 25){
    sprintf(chistg,"h1005");
    sprintf(chista,"h1105");
    sprintf(chistasv,"h1205");
```



```
    sprintf(ctext,"T_{e} = %.1f MeV",ce[5]);
  }else if(theERange == 35){
    sprintf(chistg,"h1006");
    sprintf(chista,"h1106");
    sprintf(chistasv,"h1206");
    sprintf(ctext,"T_{e} = %.1f MeV",ce[6]);
  }else if(theERange == 45){
    sprintf(chistg,"h1007");
    sprintf(chista,"h1107");
    sprintf(chistasv,"h1207");
    sprintf(ctext,"T_{e} = %.1f MeV",ce[7]);
  }else if(theERange == 55){
    sprintf(chistg,"h1008");
    sprintf(chista,"h1108");
    sprintf(chistasv,"h1208");
    sprintf(ctext,"T_{e} = %.1f MeV",ce[8]);
  }else if(theERange == 75){
    sprintf(chistg,"h1009");
    sprintf(chista,"h1109");
    sprintf(chistasv,"h1209");
    sprintf(ctext,"T_{e} = %.1f MeV",ce[9]);
  }else if(theERange == 100){
    sprintf(chistg,"h1010");
    sprintf(chista,"h1110");
    sprintf(chistasv,"h1210");
    sprintf(ctext,"T_{e} = %.1f MeV",ce[10]);
  }else if(theERange == 125){
    sprintf(chistg,"h1011");
    sprintf(chista,"h1111");
    sprintf(chistasv,"h1211");
    sprintf(ctext,"T_{e} = %.1f MeV",ce[11]);
  }else if(theERange == 150){
    sprintf(chistg,"h1012");
    sprintf(chista,"h1112");
    sprintf(chistasv,"h1212");
    sprintf(ctext,"T_{e} = %.1f MeV",ce[12]);
  }else if(theERange == 200){
    sprintf(chistg,"h1013");
    sprintf(chista,"h1113");
    sprintf(chistasv,"h1213");
    sprintf(ctext,"T_{e} = %.1f MeV",ce[13]);
  }else if(theERange == 1.5){
    sprintf(chistg,"h1014");
    sprintf(chista,"h1114");
    sprintf(chistasv,"h1214");
    sprintf(ctext,"T_{e} = %.1f MeV",ce[14]);
  }

  trtheta = new TLatex(0.50,0.80,ctext);
  trtheta->SetNDC();
  trtheta->SetTextSize(0.055);

  for (int i=0; i<3; i++) {
    int ihist = i;
    sprintf(chist2,"h%d",ihist);
    int ihist = 3+i;
    sprintf(chist,"h%d",ihist);
    switch(i) {
    case 0: hrtheta2d[i] = (TH2D*) f->Get(chistg)->Clone(chist2); break;
    case 1: hrtheta2d[i] = (TH2D*) f->Get(chista)->Clone(chist2); break;
    case 2: hrtheta2d[i] = (TH2D*) f->Get(chistasv)->Clone(chist2); break;
    default: break;
    }
    hrtheta[i] = hrtheta2d[i]->ProjectionY(chist,0,-1,"");
    //    hrtheta[i] = (TH1D*) hr->Clone(chist);
    hrtheta[i]->GetYaxis()->SetTitleOffset(1.0);
    hrtheta[i]->GetYaxis()->SetLabelSize(0.060);
    hrtheta[i]->GetYaxis()->SetLabelOffset(0.010);
    hrtheta[i]->GetXaxis()->SetLabelSize(0.060);
```

```
    hrtheta[i]->GetXaxis()->SetNdivisions(505);
    hrtheta[i]->GetXaxis()->SetLabelOffset(0.020);
    if (i==0) hrtheta[i]->GetYaxis()->SetTitle("gen");
    else hrtheta[i]->GetYaxis()->SetTitle("cm^{2}-sr");
    hrtheta[i]->GetXaxis()->CenterTitle();
    hrtheta[i]->GetXaxis()->SetTitleSize(0.070);
    hrtheta[i]->GetXaxis()->SetTitleOffset(1.2);
    hrtheta[i]->GetXaxis()->SetTitle("#theta (degree)");
    hrtheta[i]->GetYaxis()->CenterTitle();
    hrtheta[i]->GetYaxis()->SetTitleSize(0.070);
    hrtheta[i]->GetYaxis()->SetTitleOffset(1.2);
    hrtheta[i]->SetLineColor(kBlack);
    Double_t maxval = hrtheta[i]->GetMaximum();
    printf("maxval %f\n",1.25*maxval);
    hrtheta[i]->SetMaximum(1.25*maxval);
  }

  for (int i=1; i<3; i++) {
    int nbins = hrtheta[i]->GetNbinsX();
    float acc = 0;
    for (int b=1; b<=nbins; b++)
      acc+=hrtheta[i]->GetBinContent(b);
    Double_t moyen = hrtheta[i]->GetMean();
    Int_t maxbin = hrtheta[i]->GetMaximumBin();
    Double_t pic = hrtheta[i]->GetBinCenter(maxbin);
    Double_t rms = hrtheta[i]->GetRMS();
    printf("acceptance %.1f cm2-sr moyen %.1f rms %.1f pic %1.f\n",acc,moyen,rms
,pic);
  }

  gStyle->SetOptTitle(0);
  gStyle->SetOptStat(0);
  gStyle->SetCanvasColor(10);
  gStyle->SetPadColor(10);
  gStyle->SetPalette(1,0);

  TPad *ptpad;
  char cpad[30];
  TCanvas* c0 = new TCanvas("c0","theta",0,0,600,600);
  c0->SetFillColor(0);
  c0->Divide(2,2,0.001,0.001);
  for (int i=0; i<3; i++) {
    sprintf(cpad,"c0_%d",i+1);
    ptpad = (TPad*) c0->FindObject(cpad);
    ptpad->SetFillColor(10);
    ptpad->SetLogy(0);
    ptpad->SetLeftMargin(0.20);
    ptpad->SetBottomMargin(0.20);
    ptpad->SetRightMargin(0.10);
    ptpad->SetTopMargin(0.10);
    c0->cd(i+1);
    printf("c0 %d\n",i+1);
    hrtheta[i]->Draw("hist");
    trtheta->Draw();
  }


return;
}

void rtheta_p(float ec) {

  float theERange = ec;
  char chfich[100];
  sprintf(chfich,"HistProton_acc_%3.1f_MeV.root",ec);
  printf("chfich %s\n",chfich);
  TFile *f = new TFile(chfich);
  f->ls();
```



```
  char chist[60];
  char chist2[60];
  char ctext[60];
  float ce[11] = {30.0, 35.0, 55.0, 75.0, 100.0, 125.0, 150.0, 200.0, 225.0, 250
.0, 300.0};

  TH2D *hrtheta2d[3];
  TH1D *hproj;
  TH1D *hrtheta[3];
  TLatex *trtheta;

  char chistg[10];
  char chista[10];
  char chistasv[10];

  if(theERange == 30){
    sprintf(chistg,"h1000");
    sprintf(chista,"h1100");
    sprintf(chistasv,"h1200");
    sprintf(ctext,"T_{p} = %.1f MeV",ce[0]);
  }else if(theERange == 35){
    sprintf(chistg,"h1001");
    sprintf(chista,"h1101");
    sprintf(chistasv,"h1201");
    sprintf(ctext,"T_{p} = %.1f MeV",ce[1]);
  }else if(theERange == 55){
    sprintf(chistg,"h1002");
    sprintf(chista,"h1102");
    sprintf(chistasv,"h1202");
    sprintf(ctext,"T_{p} = %.1f MeV",ce[2]);
  }else if(theERange == 75){
    sprintf(chistg,"h1003");
    sprintf(chista,"h1103");
    sprintf(chistasv,"h1203");
    sprintf(ctext,"T_{p} = %.1f MeV",ce[3]);
  }else if(theERange == 100){
    sprintf(chistg,"h1004");
    sprintf(chista,"h1104");
    sprintf(chistasv,"h1204");
    sprintf(ctext,"T_{p} = %.1f MeV",ce[4]);
  }else if(theERange == 125){
    sprintf(chistg,"h1005");
    sprintf(chista,"h1105");
    sprintf(chistasv,"h1205");
    sprintf(ctext,"T_{p} = %.1f MeV",ce[5]);
  }else if(theERange == 150){
    sprintf(chistg,"h1006");
    sprintf(chista,"h1106");
    sprintf(chistasv,"h1206");
    sprintf(ctext,"T_{p} = %.1f MeV",ce[6]);
  }else if(theERange == 200){
    sprintf(chistg,"h1007");
    sprintf(chista,"h1107");
    sprintf(chistasv,"h1207");
    sprintf(ctext,"T_{p} = %.1f MeV",ce[7]);
  }else if(theERange == 225){
    sprintf(chistg,"h1008");
    sprintf(chista,"h1108");
    sprintf(chistasv,"h1208");
    sprintf(ctext,"T_{p} = %.1f MeV",ce[8]);
  }else if(theERange == 250){
    sprintf(chistg,"h1009");
    sprintf(chista,"h1109");
    sprintf(chistasv,"h1209");
    sprintf(ctext,"T_{p} = %.1f MeV",ce[9]);
  }else if(theERange == 300){
    sprintf(chistg,"h1010");
    sprintf(chista,"h1110");
    sprintf(chistasv,"h1210");
```

```
    sprintf(ctext,"T_{p} = %.1f MeV",ce[10]);
  }

  trtheta = new TLatex(0.50,0.80,ctext);
  trtheta->SetNDC();
  trtheta->SetTextSize(0.055);

  for (int i=0; i<3; i++) {
    int ihist = i;
    sprintf(chist2,"h%d",ihist);
    int ihist = 3+i;
    sprintf(chist,"h%d",ihist);
    switch(i) {
    case 0: hrtheta2d[i] = (TH2D*) f->Get(chistg)->Clone(chist2); break;
    case 1: hrtheta2d[i] = (TH2D*) f->Get(chista)->Clone(chist2); break;
    case 2: hrtheta2d[i] = (TH2D*) f->Get(chistasv)->Clone(chist2); break;
    default: break;
    }
    hrtheta[i] = hrtheta2d[i]->ProjectionY(chist,0,-1,"");
    //    hrtheta[i] = (TH1D*) hr->Clone(chist);
    hrtheta[i]->GetYaxis()->SetTitleOffset(1.0);
    hrtheta[i]->GetYaxis()->SetLabelSize(0.060);
    hrtheta[i]->GetYaxis()->SetLabelOffset(0.010);
    hrtheta[i]->GetXaxis()->SetLabelSize(0.060);
    hrtheta[i]->GetXaxis()->SetNdivisions(505);
    hrtheta[i]->GetXaxis()->SetLabelOffset(0.020);
    if (i==0) hrtheta[i]->GetYaxis()->SetTitle("gen");
    else hrtheta[i]->GetYaxis()->SetTitle("cm^{2}-sr");
    hrtheta[i]->GetXaxis()->CenterTitle();
    hrtheta[i]->GetXaxis()->SetTitleSize(0.070);
    hrtheta[i]->GetXaxis()->SetTitleOffset(1.2);
    hrtheta[i]->GetXaxis()->SetTitle("#theta (degree)");
    hrtheta[i]->GetYaxis()->CenterTitle();
    hrtheta[i]->GetYaxis()->SetTitleSize(0.070);
    hrtheta[i]->GetYaxis()->SetTitleOffset(1.2);
    hrtheta[i]->SetLineColor(kBlack);
    Double_t maxval = hrtheta[i]->GetMaximum();
    printf("maxval %f\n",1.25*maxval);
    hrtheta[i]->SetMaximum(1.25*maxval);
  }

  for (int i=1; i<3; i++) {
    int nbins = hrtheta[i]->GetNbinsX();
    float acc = 0;
    for (int b=1; b<=nbins; b++)
      acc+=hrtheta[i]->GetBinContent(b);
    Double_t moyen = hrtheta[i]->GetMean();
    Int_t maxbin = hrtheta[i]->GetMaximumBin();
    Double_t pic = hrtheta[i]->GetBinCenter(maxbin);
    Double_t rms = hrtheta[i]->GetRMS();
    printf("acceptance %.1f cm2-sr moyen %.1f rms %.1f pic %1.f\n",acc,moyen,rms
,pic);
  }

  gStyle->SetOptTitle(0);
  gStyle->SetOptStat(0);
  gStyle->SetCanvasColor(10);
  gStyle->SetPadColor(10);
  gStyle->SetPalette(1,0);

  TPad *ptpad;
  char cpad[30];
  TCanvas* c0 = new TCanvas("c0","theta",0,0,600,600);
  c0->SetFillColor(0);
  c0->Divide(2,2,0.001,0.001);
  for (int i=0; i<3; i++) {
    sprintf(cpad,"c0_%d",i+1);
    ptpad = (TPad*) c0->FindObject(cpad);
    ptpad->SetFillColor(10);
```



```
    ptpad->SetLogy(0);
    ptpad->SetLeftMargin(0.20);
    ptpad->SetBottomMargin(0.20);
    ptpad->SetRightMargin(0.10);
    ptpad->SetTopMargin(0.10);
    c0->cd(i+1);
    printf("c0 %d\n",i+1);
    hrtheta[i]->Draw("hist");
    trtheta->Draw();
  }

return;
}

void rtheta_e_bin() {

  float ce[14] = {2.0, 3.0, 5.0, 10.0, 15.0, 25.0, 35.0, 45.0, 55.0, 75.0, 100.0
,
                  125.0, 150.0, 200.0};
  int plot_bin[9] = {0, 1, 2, 3, 4, 5, 7, 9, 11};
  char cfhist[100];
  char chist[10];
  char chist2[10];
  char ctext[100];
  char chistg[10];
  char chista[10];
  char chistasv[10];
  TFile *f[14];
  TH2D *hrtheta2d[14][3];
  TH1D *hproj;
  TH1D *hrtheta[14][3];
  float hrtheta_max[14][3];
  TLatex *trtheta[14];
  int hid;

  for (int j=0; j<=13; j++) {
    sprintf(cfhist,"HistElectron_acc_%3.1f_MeV.root",ce[j]);
    printf(" %d %s\n",j,cfhist);
    f[j] = new TFile(cfhist);
    f[j]->ls();
    hid = 1000+j;
    sprintf(chistg,"h%d",hid);
    hid+=100;
    sprintf(chista,"h%d",hid);
    hid+=100;
    sprintf(chistasv,"h%d",hid);
    for (int i=0; i<3; i++) {
      int ihist = i+j*6;
      sprintf(chist2,"h%d",ihist);
      int ihist = 3+i+j*6;
      sprintf(chist,"h%d",ihist);
      switch(i) {
      case 0: hrtheta2d[j][i] = (TH2D*) f[j]->Get(chistg)->Clone(chist2); break;
      case 1: hrtheta2d[j][i] = (TH2D*) f[j]->Get(chista)->Clone(chist2); break;
      case 2: hrtheta2d[j][i] = (TH2D*) f[j]->Get(chistasv)->Clone(chist2); brea
k;
      default: break;
      }
      hrtheta[j][i] = hrtheta2d[j][i]->ProjectionY(chist,0,-1,"");
    //    hrtheta[i] = (TH1D*) hr->Clone(chist);
      hrtheta[j][i]->GetYaxis()->SetTitleOffset(1.0);
      hrtheta[j][i]->GetYaxis()->SetLabelSize(0.060);
      hrtheta[j][i]->GetYaxis()->SetLabelOffset(0.010);
      hrtheta[j][i]->GetXaxis()->SetLabelSize(0.060);
      hrtheta[j][i]->GetXaxis()->SetNdivisions(505);
      hrtheta[j][i]->GetXaxis()->SetLabelOffset(0.020);
      if (i==0) hrtheta[j][i]->GetYaxis()->SetTitle("gen");
      else hrtheta[j][i]->GetYaxis()->SetTitle("cm^{2}-sr");
      hrtheta[j][i]->GetXaxis()->CenterTitle();
```

```
      hrtheta[j][i]->GetXaxis()->SetTitleSize(0.070);
      hrtheta[j][i]->GetXaxis()->SetTitleOffset(1.2);
      hrtheta[j][i]->GetXaxis()->SetTitle("#theta (degree)");
      hrtheta[j][i]->GetYaxis()->CenterTitle();
      hrtheta[j][i]->GetYaxis()->SetTitleSize(0.070);
      hrtheta[j][i]->GetYaxis()->SetTitleOffset(1.2);
      hrtheta[j][i]->SetLineColor(kBlue);
      if (i==1 ) hrtheta[j][i]->SetLineStyle(2);
      Double_t maxval = hrtheta[j][i]->GetMaximum();
      printf("maxval %f\n",1.25*maxval);
      hrtheta_max[j][i] = maxval;
      hrtheta[j][i]->SetMaximum(1.25*maxval);
    }
    for (int i=1; i<3; i++) {
      int nbins = hrtheta[j][i]->GetNbinsX();
      float acc = 0;
      for (int b=1; b<=nbins; b++)
        acc+=hrtheta[j][i]->GetBinContent(b);
      Double_t moyen = hrtheta[j][i]->GetMean();
      Int_t maxbin = hrtheta[j][i]->GetMaximumBin();
      Double_t pic = hrtheta[j][i]->GetBinCenter(maxbin);
      Double_t rms = hrtheta[j][i]->GetRMS();
      printf("ec %f acc %.1f cm2-sr moyen %.1f rms %.1f pic %1.f\n",ce[j],acc,mo
yen,rms,pic);
    }
    sprintf(ctext,"T_{e} = %.1f MeV",ce[j]);
    trtheta[j] = new TLatex(0.50,0.80,ctext);
    trtheta[j]->SetNDC();
    trtheta[j]->SetTextSize(0.055);
  }

  gStyle->SetOptTitle(0);
  gStyle->SetOptStat(0);
  gStyle->SetCanvasColor(10);
  gStyle->SetPadColor(10);
  gStyle->SetPalette(1,0);

  TPad *ptpad;
  char cpad[30];
  TCanvas* c0 = new TCanvas("c0","theta",0,0,600,600);
  c0->SetFillColor(0);
  c0->Divide(3,3,0.001,0.001);
  for (int i=0; i<9; i++) {
    sprintf(cpad,"c0_%d",i+1);
    ptpad = (TPad*) c0->FindObject(cpad);
    ptpad->SetFillColor(10);
    ptpad->SetLogy(0);
    ptpad->SetLeftMargin(0.20);
    ptpad->SetBottomMargin(0.20);
    ptpad->SetRightMargin(0.10);
    ptpad->SetTopMargin(0.10);
    c0->cd(i+1);
    printf("c0 %d\n",i+1);
    int k = plot_bin[i];
    printf("k %d\n",k);
    if (hrtheta_max[k][2] > hrtheta_max[k][1]) {
      hrtheta[k][2]->Draw();
      hrtheta[k][1]->Draw("samehist");
    }
    else {
      hrtheta[k][1]->Draw();
      hrtheta[k][2]->Draw("samehist");
    }
    trtheta[k]->Draw();
  }

  TFile *f2 = new TFile("HistE_acc_ang_e_lyso_g3.root");
  f2->ls();
  TH1D *hrtheta_g3[10];
```



```
  for (int i=0; i<10; i++) {
    int ihist = 171 + i;
      sprintf(chist,"h%d",ihist);
      int ihist = 271 + i;
      sprintf(chist2,"h%d",ihist);
      hrtheta_g3[i] = (TH1D*) f2->Get(chist)->Clone(chist2);
      hrtheta_g3[i]->SetLineStyle(2);
  }

  TCanvas* c1 = new TCanvas("c1","theta g3/g4",0,0,600,600);
  c1->SetFillColor(0);
  c1->Divide(3,3,0.001,0.001);
  for (int i=0; i<9; i++) {
    sprintf(cpad,"c1_%d",i+1);
    ptpad = (TPad*) c1->FindObject(cpad);
    ptpad->SetFillColor(10);
    ptpad->SetLogy(0);
    ptpad->SetLeftMargin(0.20);
    ptpad->SetBottomMargin(0.20);
    ptpad->SetRightMargin(0.10);
    ptpad->SetTopMargin(0.10);
    c1->cd(i+1);
    printf("c0 %d\n",i+1);
    if (i<9) {
      hrtheta[i+2][2]->Draw();
      hrtheta_g3[i+1]->Draw("samehist");
    }
    else {
      hrtheta_g3[i+1]->Draw();
      hrtheta[i+2][2]->Draw("samehist");
    }
    trtheta[i+2]->Draw();
  }

  return;
}

void rtheta_p_bin() {

  float ce[11] = {30.0, 35.0, 55.0, 75.0, 100.0, 125.0, 150.0, 200.0, 225.0, 250
.0, 300.0};
  char cfhist[100];
  char chist[10];
  char chist2[10];
  char ctext[100];
  char chistg[10];
  char chista[10];
  char chistasv[10];
  TFile *f[14];
  TH2D *hrtheta2d[14][3];
  TH1D *hproj;
  TH1D *hrtheta[14][3];
  float hrtheta_max[14][3];
  TLatex *trtheta[14];
  int hid;

  for (int j=0; j<=10; j++) {
    sprintf(cfhist,"HistProton_acc_%3.1f_MeV.root",ce[j]);
    printf(" %d %s\n",j,cfhist);
    f[j] = new TFile(cfhist);
    f[j]->ls();
    hid = 1000+j;
    sprintf(chistg,"h%d",hid);
    hid+=100;
    sprintf(chista,"h%d",hid);
    hid+=100;
    sprintf(chistasv,"h%d",hid);
    for (int i=0; i<3; i++) {
```

```
      int ihist = i+j*6;
      sprintf(chist2,"h%d",ihist);
      int ihist = 3+i+j*6;
      sprintf(chist,"h%d",ihist);
      switch(i) {
      case 0: hrtheta2d[j][i] = (TH2D*) f[j]->Get(chistg)->Clone(chist2); break;
      case 1: hrtheta2d[j][i] = (TH2D*) f[j]->Get(chista)->Clone(chist2); break;
      case 2: hrtheta2d[j][i] = (TH2D*) f[j]->Get(chistasv)->Clone(chist2); brea
k;
      default: break;
      }
      hrtheta[j][i] = hrtheta2d[j][i]->ProjectionY(chist,0,-1,"");
    //    hrtheta[i] = (TH1D*) hr->Clone(chist);
      hrtheta[j][i]->GetYaxis()->SetTitleOffset(1.0);
      hrtheta[j][i]->GetYaxis()->SetLabelSize(0.060);
      hrtheta[j][i]->GetYaxis()->SetLabelOffset(0.010);
      hrtheta[j][i]->GetXaxis()->SetLabelSize(0.060);
      hrtheta[j][i]->GetXaxis()->SetNdivisions(505);
      hrtheta[j][i]->GetXaxis()->SetLabelOffset(0.020);
      if (i==0) hrtheta[j][i]->GetYaxis()->SetTitle("gen");
      else hrtheta[j][i]->GetYaxis()->SetTitle("cm^{2}-sr");
      hrtheta[j][i]->GetXaxis()->CenterTitle();
      hrtheta[j][i]->GetXaxis()->SetTitleSize(0.070);
      hrtheta[j][i]->GetXaxis()->SetTitleOffset(1.2);
      hrtheta[j][i]->GetXaxis()->SetTitle("#theta (degree)");
      hrtheta[j][i]->GetYaxis()->CenterTitle();
      hrtheta[j][i]->GetYaxis()->SetTitleSize(0.070);
      hrtheta[j][i]->GetYaxis()->SetTitleOffset(1.2);
      hrtheta[j][i]->SetLineColor(kRed);
      if (i==1 ) hrtheta[j][i]->SetLineStyle(2);
      Double_t maxval = hrtheta[j][i]->GetMaximum();
      printf("maxval %f\n",1.25*maxval);
      hrtheta_max[j][i] = maxval;
      hrtheta[j][i]->SetMaximum(1.25*maxval);
    }
    for (int i=1; i<3; i++) {
      int nbins = hrtheta[j][i]->GetNbinsX();
      float acc = 0;
      for (int b=1; b<=nbins; b++)
        acc+=hrtheta[j][i]->GetBinContent(b);
      Double_t moyen = hrtheta[j][i]->GetMean();
      Int_t maxbin = hrtheta[j][i]->GetMaximumBin();
      Double_t pic = hrtheta[j][i]->GetBinCenter(maxbin);
      Double_t rms = hrtheta[j][i]->GetRMS();
      printf("ec %f acc %.1f cm2-sr moyen %.1f rms %.1f pic %1.f\n",ce[j],acc,mo
yen,rms,pic);
    }
    sprintf(ctext,"T_{p} = %.1f MeV",ce[j]);
    trtheta[j] = new TLatex(0.50,0.80,ctext);
    trtheta[j]->SetNDC();
    trtheta[j]->SetTextSize(0.055);
  }

  gStyle->SetOptTitle(0);
  gStyle->SetOptStat(0);
  gStyle->SetCanvasColor(10);
  gStyle->SetPadColor(10);
  gStyle->SetPalette(1,0);

  TPad *ptpad;
  char cpad[30];
  TCanvas* c0 = new TCanvas("c0","theta",0,0,600,600);
  c0->SetFillColor(0);
  c0->Divide(3,3,0.001,0.001);
  for (int i=0; i<9; i++) {
    sprintf(cpad,"c0_%d",i+1);
    ptpad = (TPad*) c0->FindObject(cpad);
    ptpad->SetFillColor(10);
    ptpad->SetLogy(0);
```



```
    ptpad->SetLeftMargin(0.20);
    ptpad->SetBottomMargin(0.20);
    ptpad->SetRightMargin(0.10);
    ptpad->SetTopMargin(0.10);
    c0->cd(i+1);
    printf("c0 %d\n",i+1);
    if (hrtheta_max[i][2] > hrtheta_max[i][1]) {
      hrtheta[i][2]->Draw();
      hrtheta[i][1]->Draw("samehist");
    }
    else {
      hrtheta[i][1]->Draw();
      hrtheta[i][2]->Draw("samehist");
    }
    trtheta[i]->Draw();
  }

  TFile *f2 = new TFile("HistE_acc_ang_p_lyso_g3.root");
  f2->ls();
  TH1D *hrtheta_g3[10];

  for (int i=0; i<10; i++) {
    int ihist = 131 + i;
      sprintf(chist,"h%d",ihist);
      int ihist = 231 + i;
      sprintf(chist2,"h%d",ihist);
      hrtheta_g3[i] = (TH1D*) f2->Get(chist)->Clone(chist2);
      hrtheta_g3[i]->SetLineStyle(2);
  }

  TCanvas* c1 = new TCanvas("c1","theta g3/g4",0,0,600,600);
  c1->SetFillColor(0);
  c1->Divide(3,3,0.001,0.001);
  for (int i=0; i<8; i++) {
    sprintf(cpad,"c1_%d",i+1);
    ptpad = (TPad*) c1->FindObject(cpad);
    ptpad->SetFillColor(10);
    ptpad->SetLogy(0);
    ptpad->SetLeftMargin(0.20);
    ptpad->SetBottomMargin(0.20);
    ptpad->SetRightMargin(0.10);
    ptpad->SetTopMargin(0.10);
    c1->cd(i+1);
    printf("c0 %d\n",i+1);
    if (i<9) {
      hrtheta[i][2]->Draw();
      hrtheta_g3[i+2]->Draw("samehist");
    }
    else {
      hrtheta_g3[i+2]->Draw();
      hrtheta[i][2]->Draw("samehist");
    }
    trtheta[i]->Draw();
  }

  return;
}

Double_t flux_norm_e = 2383.9;
Double_t flux_norm_p = 1809.9;
Double_t parig_e[4] = { 6.83243, 22.1979, 279.525, 1.};
Double_t parig_p[4] = { 23.8664, 269.455, 109.496, 1.};

Double_t invgau(Double_t *x, Double_t *par)
{
   Double_t arg = 0;
   Double_t arg2 = 0;
   //   printf(" par %lf %lf %lf %lf\n",par[0],par[1],par[2],par[3]);
   Double_t value = 0.;
```

```
   //   x[0] += par[3];
   // mu - par[2], lambda - par[1]
   if (par[1]*par[2] != 0 && x[0] > 0.) {
      arg = (x[0] - par[2])/par[2];
      arg2 = TMath::Sqrt(par[1]/(2*TMath::Pi()*x[0]*x[0]*x[0]));
      value = par[0]*arg2*TMath::Exp(-0.5*par[1]*arg*arg/x[0]);
   }
   //   printf("arg %lf arg2 %lf\n",arg,arg2);

   //   printf("x %lf y %lf\n",x[0],par[3]*value);
   return(par[3]*value);
}

Double_t invgau_fit(Double_t *x, Double_t *par)
{
   Double_t arg = 0;
   Double_t arg2 = 0;
   //   printf(" par %lf %lf %lf %lf\n",par[0],par[1],par[2],par[3]);
   Double_t value = 0.;
   //   x[0] += par[3];
   // mu - par[2], lambda - par[1]
   if (par[1]*par[2] != 0 && x[0] > 0.) {
      arg = (x[0] - par[2])/par[2];
      arg2 = TMath::Sqrt(par[1]/(2*TMath::Pi()*x[0]*x[0]*x[0]));
      value = par[0]*arg2*TMath::Exp(-0.5*par[1]*arg*arg/x[0]);
   }
   //   printf("arg %lf arg2 %lf\n",arg,arg2);

   //   printf("x %lf y %lf\n",x[0],par[3]*value);
   return(value);
}

void acc_ep_ccf4l4() {
  // MeV bin2/bin3
  float ece[13] = {2.5, 5.0, 6.7, 8.4, 10.0, 15.0, 25.0, 35.0, 45.0, 55.0, 65.0,
 75.0, 100.0};
  // cm2-sr
  float eacc_sci_cf4_l4[13] = {129.1., 321.0, 347.5, 350.9, 339.6, 285.6, 186.4,
 123.0, 87.4, 68.4, 57.3, 50.0, 37.7};
  // MeV bin4
  float pce[10] = {22.0, 25.0, 30.0, 35.0, 55.0, 75.0, 100.0, 125.0, 150.0, 200.
0};
  // cm2-sr
  float pacc_sci_cf4_l4[10] = {0.0, 0.0, 139.0, 250.5, 380.7, 357.4, 279.7, 200.
8, 135.5, 66.3};
  float x[15],y[15];

  TF1 *invg_e = new TF1("invg_e",invgau,0,120,4);
  invg_e->SetLineWidth(1);
  invg_e->SetLineColor(kBlue);
  invg_e->SetLineStyle(1);

  TF1 *invg_p = new TF1("invg_p",invgau,20,200,4);
  invg_p->SetLineWidth(1);
  invg_p->SetLineColor(kRed);
  invg_p->SetLineStyle(1);

  TF1 *invg_e_fit = new TF1("invg_e_fit",invgau_fit,0,120,3);
  invg_e_fit->SetLineWidth(1);
  invg_e_fit->SetLineColor(kBlue);
  invg_e_fit->SetLineStyle(1);

  TF1 *invg_p_fit = new TF1("invg_p_fit",invgau_fit,20,200,3);
  invg_p_fit->SetLineWidth(1);
  invg_p_fit->SetLineColor(kRed);
  invg_p_fit->SetLineStyle(1);

  TH1F *haccfite = new TH1F("haccfit"," ",201,0.,210.);
  haccfite->SetMaximum(.5);
```



```
haccfite->SetMinimum(0.01);
haccfite->GetYaxis()->SetTitleOffset(1.0);
haccfite->GetYaxis()->SetLabelSize(0.045);
haccfite->GetYaxis()->SetLabelOffset(0.01);
haccfite->GetXaxis()->SetLabelSize(0.050);
//  hacc->GetXaxis()->SetNdivisions(505);
haccfite->GetXaxis()->SetLabelOffset(0.005);
haccfite->GetYaxis()->SetTitle("cm^{2}-sr");
haccfite->GetXaxis()->CenterTitle();
haccfite->GetXaxis()->SetTitleSize(0.055);
haccfite->GetXaxis()->SetTitleOffset(1.3);
haccfite->GetXaxis()->SetTitle("Kinetic Energy (MeV)");
haccfite->GetYaxis()->CenterTitle();
haccfite->GetYaxis()->SetTitleSize(0.055);
haccfite->GetYaxis()->SetTitleOffset(1.5);

Double_t tacce = 0;
Double_t taccp = 0;
for (int i=0; i<13; i++) {
  tacce += eacc_sci_cf4_l4[i];
}
printf("tacce %.2f \n",tacce);
for (int i=0; i<10; i++) {
  taccp += pacc_sci_cf4_l4[i];
}
printf("taccp %.2f\n",taccp);
parig_e[3] = tacce;
parig_p[3] = taccp;

int np = 0;
for (int i=0; i<13; i++) {
  x[np] = ece[i];
  y[np] = eacc_sci_cf4_l4[i]/tacce;
  haccfite->Fill(ece[i],eacc_sci_cf4_l4[i]/tacce);
  np++;
}

TGraph *gacc_esci_cf4_l4_fit = new TGraph(np,x,y);
Double_t parig[4] = {1., 5.0, 10.0, 1.};
invg_e_fit->SetParameters(parig);
gacc_esci_cf4_l4_fit->Fit("invg_e_fit","R");

for (int i=0; i<12; i++) {
  //    pacc_sci_cf4_l4[i] /= taccp;
  //    haccfite->Fill(pce[i],pacc_sci_cf4_l4[i]);
}

gStyle->SetOptTitle(0);
gStyle->SetOptStat(0);
gStyle->SetCanvasColor(10);
gStyle->SetPadColor(10);
gStyle->SetPalette(1,0);

TPad *ptpad;
char cpad[30];
TCanvas* c0 = new TCanvas("c0","accfite",0,0,500,500);
c0->SetFillColor(0);
c0->Divide(1,1,0.001,0.001);
sprintf(cpad,"c0_%d",1);
ptpad = (TPad*) c0->FindObject(cpad);
ptpad->SetFillColor(10);
ptpad->SetLogy(0);
ptpad->SetLogx(0);
ptpad->SetLeftMargin(0.20);
ptpad->SetBottomMargin(0.15);
ptpad->SetRightMargin(0.15);
ptpad->SetTopMargin(0.150);
ptpad->SetTickx(1);
ptpad->SetTicky(1);
```

```
c0->cd(1);
haccfite->Draw();
//  invg_e_fit->Draw("same");

//  return;
int np = 0;
for (int i=0; i<13; i++) {
  x[np] = ece[i];
  y[np] = eacc_sci_cf4_l4[i];
  np++;
}

TGraph *gacc_esci_cf4_l4 = new TGraph(np,x,y);
gacc_esci_cf4_l4->SetMarkerStyle(20);
gacc_esci_cf4_l4->SetMarkerColor(kBlue);
gacc_esci_cf4_l4->SetMarkerSize(0.8);
gacc_esci_cf4_l4->SetLineWidth(1);
gacc_esci_cf4_l4->SetLineStyle(1);
gacc_esci_cf4_l4->SetLineColor(kBlue);
Double_t parig[4] = {1., 2.0, 2.0, 1.};
invg_e_fit->SetParameters(parig);
//  gacc_esci_cf4_l4->Fit("invg_e_fit","R");

np = 0;
for (int i=0; i<10; i++) {
  if (pacc_sci_cf4_l4[i] > 0.) {
  x[np] = pce[i];
  y[np] = pacc_sci_cf4_l4[i];
  np++;
  }
}

TGraph *gacc_psci_cf4_l4 = new TGraph(np,x,y);
gacc_psci_cf4_l4->SetMarkerStyle(20);
gacc_psci_cf4_l4->SetMarkerColor(kRed);
gacc_psci_cf4_l4->SetMarkerSize(0.8);
gacc_psci_cf4_l4->SetLineWidth(2);
gacc_psci_cf4_l4->SetLineStyle(2);
gacc_psci_cf4_l4->SetLineColor(kRed);
Double_t parig[4] = {1., 5.0, 60.0, 1.0};
//  invg_p->SetParameters(parig);
//  gacc_psci_cf4_l4->Fit("invg_p","R");

//  return;

TH1F *hacc = new TH1F("hacc"," ",50,1.0,1000.);
hacc->SetMaximum(450.0);
hacc->SetMinimum(0.0);
hacc->GetYaxis()->SetTitleOffset(1.0);
hacc->GetYaxis()->SetLabelSize(0.045);
hacc->GetYaxis()->SetLabelOffset(0.01);
hacc->GetXaxis()->SetLabelSize(0.050);
//  hacc->GetXaxis()->SetNdivisions(505);
hacc->GetXaxis()->SetLabelOffset(0.005);
hacc->GetYaxis()->SetTitle("cm^{2}-sr");
hacc->GetXaxis()->CenterTitle();
hacc->GetXaxis()->SetTitleSize(0.055);
hacc->GetXaxis()->SetTitleOffset(1.3);
hacc->GetXaxis()->SetTitle("Kinetic Energy (MeV)");
hacc->GetYaxis()->CenterTitle();
hacc->GetYaxis()->SetTitleSize(0.055);
hacc->GetYaxis()->SetTitleOffset(1.5);

TCanvas* c2 = new TCanvas("c2","acc",0,0,500,500);
c2->SetFillColor(0);
c2->Divide(1,1,0.001,0.001);
sprintf(cpad,"c2_%d",1);
ptpad = (TPad*) c2->FindObject(cpad);
ptpad->SetFillColor(10);
```



```
ptpad->SetLogy(0);
ptpad->SetLogx(1);
ptpad->SetLeftMargin(0.20);
ptpad->SetBottomMargin(0.15);
ptpad->SetRightMargin(0.15);
ptpad->SetTopMargin(0.150);
ptpad->SetTickx(1);
ptpad->SetTicky(1);
c2->cd(1);
hacc->Draw();
gacc_esci_cf4_l4->Draw("P");
gacc_psci_cf4_l4->Draw("P");
invg_e->SetRange(2.5,100.);
invg_e->SetParameters(parig_e);
invg_e->Draw("same");
invg_p->SetRange(28.,200.);
invg_p->SetParameters(parig_p);
invg_p->Draw("same");

TLatex *te = new TLatex(0.275,0.760,"electrons");
te->SetNDC();
te->SetTextColor(kBlue);
te->SetTextSize(0.04);
te->Draw();

TLatex *tp = new TLatex(0.65,0.755,"protons");
tp->SetNDC();
tp->SetTextColor(kRed);
tp->SetTextSize(0.04);
tp->Draw();

x[0] = 2.;
y[0] = 38.;
x[1] = 3.5;
y[1] = 38.;
TLine *lcf1 = new TLine(x[0],y[0],x[1],y[1]);
lcf1->SetLineWidth(2);
lcf1->SetLineStyle(1);
lcf1->SetLineColor(kBlack);
//  lcf1->Draw();

TLatex *tcf1 = new TLatex(0.34,0.242,"with LYSO");
tcf1->SetNDC();
tcf1->SetTextSize(0.035);
//  tcf1->Draw();

y[0] = 66.;
y[1] = 66.;
TLine *lcf2 = new TLine(x[0],y[0],x[1],y[1]);
lcf2->SetLineWidth(2);
lcf2->SetLineStyle(2);
lcf2->SetLineColor(kBlack);
//  lcf2->Draw();

TLatex *tcf2 = new TLatex(0.34,0.202,"without LYSO");
tcf2->SetNDC();
tcf2->SetTextSize(0.035);
//  tcf2->Draw();

return;
}
```