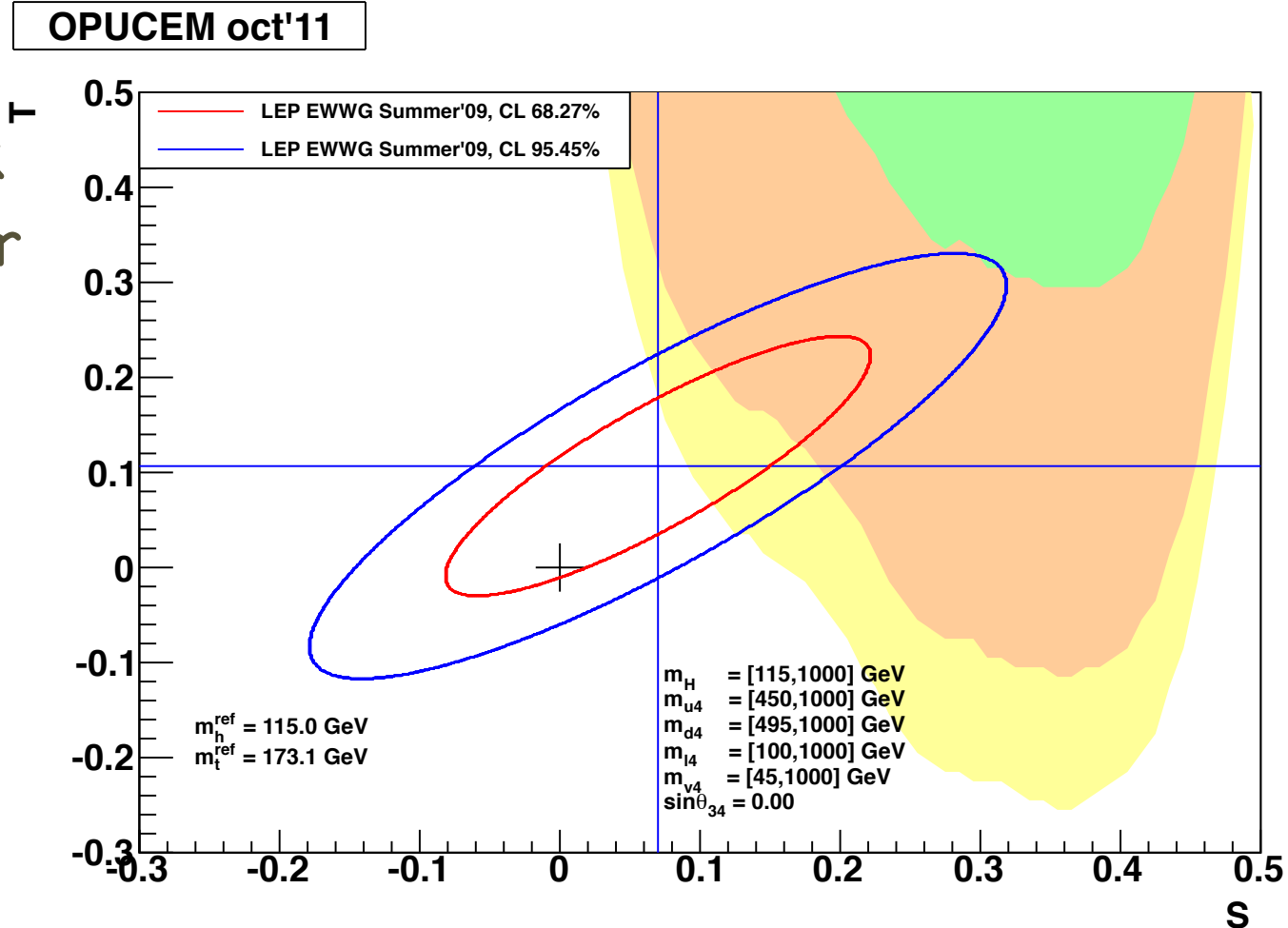


Yapılacak işler

- Daha önce hazırlanmış olan 3 root kütüğü ektedir.
 - ➔ root kütüklerinin içinde 2 boyutlu S-T uzayında sonuç veren parametre taraması ve bu taramada kullanılan parametrelerin başlangıç ve bitiş değerleri var.

- Amaç

- ➔ Makaleye konacak şekilde çizilmiş bir resim hazırlamak.



Adımlar 1

- root kütüklerinden birini yükleyip bakın, parametre değerlerini ekrana yazın

- bir root macro yapmaya başlayalım: drawST.C

- ➔ x,y başlangıç ve bitiş değerleri kullanıcı tarafından verilen bir işlev yazın, sadece çerçeve çizsin.

```
void drawST( float xmin=-0.3, float xmax=0.5, float ymin=-0.3, float
ymax=0.5 ) {
.....
}
```

- ➔ Taranan parametrelerin sınırlarını root kütüğünden okuyup resmin üzerine koyan bir işlev yazın

```
void drawSTlegends ( ) {
double h_min,u4_min,d4_min,v4_min,l4_min,s34_min,N4_min;
double h_max,u4_max,d4_max,v4_max,l4_max,s34_max,N4_max;
double mh_ref,mt_ref,S2W_ref;
.....
char tem1[32], tem2[32];
TString dum_str;
TLatex *tl = new TLatex();
          tl->SetTextSize(0.026);
          tl->SetTextAlign(12);
.....
}
```

Adımlar 2

- bir root macro yapmaya başlayalım: fig.C

➔ bu macro bir öncekini kullanarak veri kütüğünü açsın ve sınırları resme yazsın:

```
#include "drawST.C"
void fig() {
    drawST();
    ....
    drawSTlegends();
}
```

- ilk denememizi yapalım

➔ root -x fig.C

Adımlar 3

- drawST.C içinde yeni işlev tanımlayalım.

➔ bu işlev ST uzayında taranan alanın sınırlarını bulsun:

```
void drawSTarea (int shadecolor = kYellow) {  
    ....  
    int npmax=np;  
    TGraph *Shade1 = new TGraph(2*npmax);  
    TPolyMarker *pm = new TPolyMarker(npmax);  
    .....  
    gStyle->SetOptStat(0);  
    Shade1->SetFillColor(shadecolor);  
    Shade1->SetFillStyle(1001);  
    Shade1->Draw("f");  
}
```

- fig.C yi de yenileyelim

➔ yeni yazdığımız işlevi kullanalım.

```
#include "drawST.C"  
void fig() {  
    drawST();  
    .....  
    drawSTarea();  
    drawSTlegends();  
}
```

Adımlar 4

- fig.C yi de birbiri ardına 3 root kütüğü açalım
 - ➔ 3 alanı da farklı renklerle çizdirelim.

```
#include "drawST.C"
void fig() {
    drawST();
    .....
    drawSTarea(kCyan);
    .....
    drawSTarea(kYellow);
    drawSTlegends();
}
```

- drawST.C'ye son bir ekleme yapalım

- ➔ SM artısını, elipsleri çizelim ve resmi saklayalım

```
void drawSTextras( string outn="outfile.pdf" ){
    TMarker *marksm = new TMarker(0, 0, 2);
    marksm->SetMarkerSize(3); marksm->Draw();
}
```

Adımlar 5

- drawST.C'ye son bir ekleme yapalım

➔ SM artısını, S,T elipsi merkezini, elipsleri çizelim ve resmi saklayalım

```
void drawSTextras( string outn="outfile.pdf"){
    EColor col=kBlue;
    TMarker *marksm = new TMarker(0, 0, 2);
    marksm->SetMarkerSize(3); marksm->Draw();
    ...
    const double Sshift = 0.020, Tshift = 0.0067;
    const double S=0.05+Sshift, sigS=0.10, T=0.10+Tshift,
    sigT=0.09, rho=0.85;
    TF2 *f5 = drawEll(S, T, sigS, sigT, rho, 6.18,
    "EWG09_2sigma", col, false);
    col=kRed;
    TF2 *f4 = drawEll(S, T, sigS, sigT, rho, 2.30,
    "EWG09_1sigma", col);
    ...
}
```

yardımcı

- elips çizmek için yardımcı işlev

```
TF2* drawEll(double S, double T, double sigS, double sigT, double rho,
double delChi2, TString name="fell", EColor color=kBlack)
{
double angle = (sigS==sigT) ? (TMath::PiOver4()) : atan(2*rho*sigS*sigT/
(sigS*sigS-sigT*sigT))/2;
double cot = 1/tan(angle);
double b = sqrt( (pow(sigT,2)*sigS*(-1 + pow(rho,2))*(sigS*cos(2*angle)
+ sigT*rho*sin(2*angle))) / (-((pow(sigT,2)*(-1 + 2*pow(rho,2)) +
2*sigT*sigS*rho*cot + pow(sigS,2)*pow(cot,2))*pow(sin(angle),2)) ) );
double a = sqrt( pow(sin(angle),2) / (1/(1-rho*rho)/sigT/sigT -
pow(cos(angle)/b,2)) );

TF2 *fell = new TF2(name, "(x-[0])*cos([4])+(y-[1])*sin([4]))^2/[2]^2 +
((x-[0])*sin([4])-(y-[1])*cos([4]))^2/[3]^2 < [5]",-0.2, 0.4, -0.2,
0.4);

fell->SetParameters(S, T, a, b, angle, delChi2);
fell->SetNpy(500);      fell->SetNpx(500);
fell->SetLineWidth(1); fell->SetLineStyle(1); fell->SetLineColor(color);
fell->Draw("same");

return fell;
}
```

Adımlar 6 - son

- fig.C yi de yeni işlevi kullanalım

➔ Çıktıyı farklı isimle saklayalım.

```
#include "drawST.C"
void fig() {
    drawST();
    .....
    drawSTarea(kCyan);
    .....
    drawSTarea(kYellow);
    drawSTlegends();
    drawSTextras("fig1.eps");
}
```