

Jul 19, 09 14:13	Align.cxx	Page 1/17
------------------	------------------	-----------

```

#include "Align.hh"

ClassImp(Align);

Align::Align(int value, int run) {

    printf("align\n");
    align = value;
    npts = 5000;
    ntrace = 0;
    for (int i=0; i<2; i++)
        for (int j=0; j<6; j++) ntrace_eff[i][j] = 0.;
    nfit1 = 0;
    nfit2 = 0;
    nfit2_2 = 0;
    nfit3 = 0;
    nfit4 = 0;
    nfit5 = 0;
    nfit6 = 0;
    nfit7 = 0;
    nfit8 = 0;
    refl = 1;
    ref2 = 2;
    ind3 = 3;
    ind4 = 0;
    indp1 = 5;
    indp2 = 4;
    ztdr[0] = 5; // b1
    ztdr[1] = 1; // l04pt002
    ztdr[2] = 3; // l12ai009
    ztdr[3] = 0; // l04pt001
    ztdr[4] = 4; // b2
    ztdr[5] = 2; // l12ai002

    plaq_dim[0] = 7.2045;
    plaq_dim[1] = 4.1360;
    plaq_dim_active[0] = 7.0290;
    plaq_dim_active[1] = 3.9936;
    ecart = 0.0040;

    for (int i=0; i<6; i++)
        for (int j=10; j<13; j++)
            alqpar[j][i] = 0.;

    alqpar[14][0] = 75.7;
    alqpar[14][1] = 5.6;
    alqpar[14][2] = 94.5;
    alqpar[14][3] = 38.4;
    alqpar[14][4] = 85.2;
    alqpar[14][5] = 0.0;

    for (int i=0; i<6; i++)
        for (int j=10; j<13; j++)
            alqpar_2[j][i] = 0.;

    alqpar_2[14][0] = 75.7;
    alqpar_2[14][1] = 5.6;
    alqpar_2[14][2] = 94.5;
    alqpar_2[14][3] = 38.4;
    alqpar_2[14][4] = 85.2;
    alqpar_2[14][5] = 0.0;
    for (int i=0; i<6; i++) printf(" alqpar_2 14 %fi %d\n", alqpar_2[14][i], i);

    printf("align %d run %d\n", align, run);
    if (align == 1) {
        switch (run) {
            case 1225:
                alqpar[10][3] = 15.;

```

Jul 19, 09 14:13	Align.cxx	Page 2/17
------------------	------------------	-----------

```

                alqpar[11][3] = 0.5;
                break;
            case 1226:
                alqpar[10][3] = 15.;
                alqpar[11][3] = 0.5;
                break;
            case 1230:
                alqpar[10][3] = 20.;
                alqpar[11][3] = 0.5;
                break;
            case 1232:
                alqpar[10][3] = 30.;
                alqpar[11][3] = 0.5;
                break;
            case 1258:
                alqpar[10][3] = 40.;
                alqpar[11][3] = 0.5;
                break;
            case 1261:
                alqpar[10][3] = 50.;
                alqpar[11][3] = 0.5;
                break;
            case 1235:
                alqpar[12][3] = 10.;
                alqpar[13][3] = 0.5;
                break;
            case 1236:
                alqpar[12][3] = 10.;
                alqpar[13][3] = 0.5;
                break;
            case 1237:
                alqpar[12][3] = 20.;
                // alqpar[12][3] = 0.5;
                alqpar[13][3] = 0.0;
                alqpar[1][3] -= ((plaq_dim[1]-plaq_dim_active[1]) + ecart + plaq_dim_active
[1]);
                printf("alqpar[1][3] %f\n", alqpar[1][3]);
                alqpar_2[12][3] = 20.;
                alqpar_2[13][3] = 1.0;
                break;
            case 1238:
                alqpar[12][3] = 20.;
                alqpar[13][3] = 0.5;
                break;
            case 1239:
                alqpar[12][3] = 30.;
                alqpar[13][3] = 0.5;
                break;
            case 1240:
                alqpar[12][3] = 30.;
                alqpar[13][3] = 0.5;
                break;
            case 1241:
                alqpar[12][3] = 30.;
                alqpar[13][3] = 0.5;
                break;
            case 1263:
                alqpar[12][3] = 40.;
                alqpar[13][3] = 0.5;
                break;
            case 1243:
                alqpar[10][3] = 43.5;
                alqpar[11][3] = 0.5;
                alqpar[12][3] = 20.;
                alqpar[13][3] = 0.5;
                printf("case 1243\n");
                break;
            case 1244:
                alqpar[10][3] = 43.5;

```

Jul 19, 09 14:13

Align.cxx

Page 3/17

```

    alqpar[11][3] = 0.5;
    alqpar[12][3] = 20.;
    alqpar[13][3] = 0.5;
    break;
case 1246:
    alqpar[10][3] = 30.;
    alqpar[11][3] = 0.5;
    alqpar[12][3] = 30.;
    alqpar[13][3] = 0.5;
    break;
case 1247:
    alqpar[10][3] = 30.;
    alqpar[11][3] = 0.5;
    alqpar[12][3] = 30.;
    alqpar[13][3] = 0.5;
    break;
}
}

for (int i=0; i<6; i++) {
    if (i != 3) ppiste_n[i]=641.;
    else ppiste_n[i]=833.;
}

for (int i=0; i<6; i++) deux_plaq_n[i] = 0;
if (run == 1237) deux_plaq_n[3] = 1;

for (int i=0; i<6; i++) delta_y_ind4 = 0;
if (run == 1237) delta_y_ind4 = -0.027;
// if (run == 1237) delta_y_ind4 = -0.027 -0.031;

for (int i=0; i<6; i++)
    for (int j=0; j<1024; j++) mauvais[i][j]=0;

// canaux chauds du detecteur lu par TDR 2111
printf("commence mauvais\n");
// const int nc = 19;
const int nc = 26;
// int canaux[nc] = { 1, 2, 3, 4, 5, 6, 7, 128, 129, 130, 131, 133, 135, 137,
//                  195, 198, 201, 230, 235 };
int canaux[nc] = { 1, 2, 3, 4, 5, 6, 7, 128, 129, 130, 131, 132, 133, 134, 135,
136, 137,
195, 196, 197, 198, 199, 200, 201, 230, 235 };
for (int i=0; i<nc; i++)
    mauvais[5][canaux[i]-1]=1;

for(int i=0; i<6; i++) {
    resx[i] = -999.;
    resex[i] = -999.;
    reseffx[i] = -999.;
    xpospred[i] = -999.;
    resy[i] = -999.;
    reseyy[i] = -999.;
    ypospred[i] = -999.;
    xdx[i] = 0.;
    ydy[i] = 0.;
}

for(int i=0; i<4; i++) {
    resxech[i] = -999.;
    resexech[i] = -999.;
    resyech[i] = -999.;
    reseych[i] = -999.;
}

for(int i=0; i<2; i++) {
    resx_pos0[i] = -999.;
    resy_pos0[i] = -999.;
}

```

Sunday July 19, 2009

Jul 19, 09 14:13

Align.cxx

Page 4/17

```

    resx_pos4[i] = -999.;
    resy_pos4[i] = -999.;
}

for (int i=0; i<11; i++)
    for (int j=0; j<6; j++)
        for (int k=0; k<8; k++) alqpar_fich[k][i][j]=0.;

fitxz = new TLinearFitter(1,"pol1");
fitxz->StoreData(0);
fityz = new TLinearFitter(1,"pol1");
fityz->StoreData(0);
xp = new Double_t[6];
yp = new Double_t[6];
zp = new Double_t[6];
e = new Double_t[6];
}

void Align::EcrireAlgPar(){
    FILE * ft=fopen("alqpar.dat", "w");
    if(!ft){
        printf("Error cannot open \n");
        return;
    }

    for (int i=0; i<13; i+=2) {
        fprintf(ft,"%f%f%f%f%f%f\n",alqpar[i][0],alqpar[i][1],
        alqpar[i][2],alqpar[i][3],alqpar[i][4],alqpar[i][5]);
        fprintf(ft,"%f%f%f%f%f%f\n",alqpar[i+1][0],alqpar[i+1][1],
        alqpar[i+1][2],alqpar[i+1][3],alqpar[i+1][4],alqpar[i+1][5]);
    }
    fprintf(ft,"%f%f%f%f%f%f\n",alqpar[14][0],alqpar[14][1],alqpar[14][2],alqpar
[14][3],
        alqpar[14][4],alqpar[14][5]);

    if (deux_plaq()) {
        for (int i=0; i<13; i+=2) {
            fprintf(ft,"%f%f%f%f%f%f\n",alqpar_2[i][0],alqpar_2[i][1],
            alqpar_2[i][2],alqpar_2[i][3],alqpar_2[i][4],alqpar_2[i][5]);
            fprintf(ft,"%f%f%f%f%f%f\n",alqpar_2[i+1][0],alqpar_2[i+1][1],
            alqpar_2[i+1][2],alqpar_2[i+1][3],alqpar_2[i+1][4],alqpar_2[i+1][5]);
        }
        fprintf(ft,"%f%f%f%f%f%f\n",alqpar_2[14][0],alqpar_2[14][1],alqpar_2[14][2]
,alqpar_2[14][3],
            alqpar_2[14][4],alqpar_2[14][5]);
    }

    fclose(ft);
    return;
}

void Align::LireAlgPar(){

    char filename[40]="../align/alqpar.dat";
    char ligne[100];
    printf("Lecture de %s\n", filename);

    FILE * ft=fopen(filename, "r");
    if(!ft){
        printf("Error cannot open %s \n", filename);
        return;
    }

    float x00,y00,x01,y01,x02,y02,x03,y03,x04,y04,x05,y05;
    for (int i=0; i<13; i+=2) {
        fgets(ligne,sizeof(ligne),ft);

```

Align.cxx

2/9

Jul 19, 09 14:13

Align.cxx

Page 5/17

```

sscanf(ligne, "%f%f%f%f%f\n", &x00, &x01,
&x02, &x03, &x04, &x05);
algp[ar][i][0] = x00;
algp[ar][i][1] = x01;
algp[ar][i][2] = x02;
algp[ar][i][3] = x03;
algp[ar][i][4] = x04;
algp[ar][i][5] = x05;
printf("%f%f%f%f%f\n", algp[ar][i][0], algp[ar][i][1],
algp[ar][i][2], algp[ar][i][3], algp[ar][i][4], algp[ar][i][5]);
fgets(ligne, sizeof(ligne), ft);
sscanf(ligne, "%f%f%f%f%f\n", &y00, &y01, &y02,
&y03, &y04, &y05);
algp[ar][i+1][0] = y00;
algp[ar][i+1][1] = y01;
algp[ar][i+1][2] = y02;
algp[ar][i+1][3] = y03;
algp[ar][i+1][4] = y04;
algp[ar][i+1][5] = y05;
printf("%f%f%f%f%f\n", algp[ar][i+1][0], algp[ar][i+1][1],
algp[ar][i+1][2], algp[ar][i+1][3], algp[ar][i+1][4], algp[ar][i+1][5]);
}
fgets(ligne, sizeof(ligne), ft);
sscanf(ligne, "%f%f%f%f%f\n", &x00, &x01, &x02, &x03, &x04, &x05);
algp[ar][14][0] = x00;
algp[ar][14][1] = x01;
algp[ar][14][2] = x02;
algp[ar][14][3] = x03;
algp[ar][14][4] = x04;
algp[ar][14][5] = x05;
printf("%f%f%f%f%f\n", algp[ar][14][0], algp[ar][14][1], algp[ar][14][2], algp[ar][14]
[3],
algp[ar][14][4], algp[ar][14][5]);
/*
float x00,y00,x01,y01,x02,y02,x03,y03;
int ret=fscanf(ft, "%f%f%f%f%f\n", &x00, &y00, &x01, &y01, &x02, &y02,
&x03, &y03);
// printf("%d %f %f %f %f %f %f %f\n", ret, x00, y00, x01, y01, x02, y02, x03, y03);
algp[ar][0][0]=x00;
algp[ar][1][0]=y00;

algp[ar][0][1]=x01;
algp[ar][1][1]=y01;

algp[ar][0][2]=x02;
algp[ar][1][2]=y02;

algp[ar][0][3]=x03;
algp[ar][1][3]=y03;
*/

if (deux_plaq()) {
for (int i=0; i<13; i+=2) {
fgets(ligne, sizeof(ligne), ft);
sscanf(ligne, "%f%f%f%f%f\n", &x00, &x01,
&x02, &x03, &x04, &x05);
algp[ar_2][i][0] = x00;
algp[ar_2][i][1] = x01;
algp[ar_2][i][2] = x02;
algp[ar_2][i][3] = x03;
algp[ar_2][i][4] = x04;
algp[ar_2][i][5] = x05;
printf("%f%f%f%f%f\n", algp[ar_2][i][0], algp[ar_2][i][1],
algp[ar_2][i][2], algp[ar_2][i][3], algp[ar_2][i][4], algp[ar_2][i][5]);
fgets(ligne, sizeof(ligne), ft);
sscanf(ligne, "%f%f%f%f%f\n", &y00, &y01, &y02,
&y03, &y04, &y05);
algp[ar_2][i+1][0] = y00;
algp[ar_2][i+1][1] = y01;

```

Jul 19, 09 14:13

Align.cxx

Page 6/17

```

algp[ar_2][i+1][2] = y02;
algp[ar_2][i+1][3] = y03;
algp[ar_2][i+1][4] = y04;
algp[ar_2][i+1][5] = y05;
printf("%f%f%f%f%f\n", algp[ar_2][i+1][0], algp[ar_2][i+1][1],
algp[ar_2][i+1][2], algp[ar_2][i+1][3], algp[ar_2][i+1][4], algp[ar_2][i+1][5]);
}
fgets(ligne, sizeof(ligne), ft);
sscanf(ligne, "%f%f%f%f%f\n", &x00, &x01, &x02, &x03, &x04, &x05);
algp[ar_2][14][0] = x00;
algp[ar_2][14][1] = x01;
algp[ar_2][14][2] = x02;
algp[ar_2][14][3] = x03;
algp[ar_2][14][4] = x04;
algp[ar_2][14][5] = x05;
printf("%f%f%f%f%f\n", algp[ar_2][14][0], algp[ar_2][14][1], algp[ar_2][14][2], al
gpar_2[14][3],
algp[ar_2][14][4], algp[ar_2][14][5]);
}

fclose(ft);
return;
}

void Align::LireFichAlgp[ar](int run){

char almdir[255];
char algfich[255];
sprintf(almdir, ".\\align ");
if (run >= 1500 && run <= 1515)
printf(algfich, "%s/algp[ar]_d_1_2_6_etan.dat", almdir, run);
// else
// if ( run == 1261 && amaspar[0][3] == 3)
// printf(algfich, "%s/algp[ar]_d_1_2_6_3p.dat", almdir, run);
// else
// if ( run == 1261 && amaspar[0][3] == 3)
// printf(algfich, "%s/algp[ar]_d_1_2_6_3p.dat", almdir, run);
else {
for (int i=0; i<6; i++) printf("i %d npiste p %d n %d\n", i, amaspar[0][i], amaspar[1]
[i]);
if (run == 1234 || run == 1251)
printf(algfich, "%s/algp[ar]_d_1_2_6_etan.dat", almdir, run);
else {
if (amaspar[0][3] >= 3)
printf(algfich, "%s/algp[ar]_d_1_2_6_%dp_etan.dat", almdir, run, amaspar[0][3]);

// printf(algfich, "%s/algp[ar]_d_1_2_6_3p_etan.dat", almdir, run);
if (amaspar[1][3] >= 3)
printf(algfich, "%s/algp[ar]_d_1_2_6_%dp_etan3.dat", almdir, run, amaspar[1][3]);

// printf(algfich, "%s/algp[ar]_d_1_2_6_3p_etan.dat", almdir, run);
if (amaspar[0][3] >= 3 && amaspar[1][3] >= 3)
printf(algfich, "%s/algp[ar]_d_1_2_6_%dp_%dp_etan.dat", almdir, run, amaspar[0][3], a
maspar[1][3]);
// printf(algfich, "%s/algp[ar]_d_1_2_6_3p_etan.dat", almdir, run);
if (amaspar[0][3] == 2 && amaspar[1][3] == 2)
printf(algfich, "%s/algp[ar]_d_1_2_6_etan.dat", almdir, run);
}
if (run == 1246 || run == 1247) printf(algfich, "%s/algp[ar]_d_1_2_6_%dp_etan.dat",
almdir, run, amaspar[0][3]);
}
printf("algfich %s\n", algfich);
FILE *ft;
ft = fopen(algfich, "r");
if (!ft){
printf("Error cannot open %s\n", algfich);
return;
}
float x00,y00,x01,y01,x02,y02,x03,y03,x04,y04,x05,y05;

```

Jul 19, 09 14:13

Align.cxx

Page 7/17

```

char ligne[100];
for (int i=0; i<13; i+=2) {
    fgets(ligne,sizeof(ligne),ft);
    sscanf(ligne,"%f%f%f%f%f%f\n",&x00,&x01,
    &x02,&x03,&x04,&x05);
    alqpar[i][0] = x00;
    alqpar[i][1] = x01;
    alqpar[i][2] = x02;
    alqpar[i][3] = x03;
    alqpar[i][4] = x04;
    alqpar[i][5] = x05;
    printf("%f%f%f%f%f%f\n",alqpar[i][0],alqpar[i][1],
    alqpar[i][2],alqpar[i][3],alqpar[i][4],alqpar[i][5]);
    fgets(ligne,sizeof(ligne),ft);
    sscanf(ligne,"%f%f%f%f%f%f\n",&y00,&y01,&y02,
    &y03,&y04,&y05);
    alqpar[i+1][0] = y00;
    alqpar[i+1][1] = y01;
    alqpar[i+1][2] = y02;
    alqpar[i+1][3] = y03;
    alqpar[i+1][4] = y04;
    alqpar[i+1][5] = y05;
    printf("%f%f%f%f%f%f\n",alqpar[i+1][0],alqpar[i+1][1],
    alqpar[i+1][2],alqpar[i+1][3],alqpar[i+1][4],alqpar[i+1][5]);
}
fgets(ligne,sizeof(ligne),ft);
sscanf(ligne,"%f%f%f%f%f%f\n",&x00,&x01,&x02,&x03,&x04,&x05);
alqpar[14][0] = x00;
alqpar[14][1] = x01;
alqpar[14][2] = x02;
alqpar[14][3] = x03;
alqpar[14][4] = x04;
alqpar[14][5] = x05;
printf("%f%f%f%f%f%f\n",alqpar[14][0],alqpar[14][1],alqpar[14][2],alqpar[14]
[3],
    alqpar[14][4],alqpar[14][5]);
/*
float x00,y00,x01,y01,x02,y02,x03,y03;
int ret=fscanf(ft,"%f%f%f%f%f%f\n",&x00,&y00,&x01,&y01,&x02,&y02,
&x03,&y03);
// printf("%d%f%f%f%f%f%f\n",ret,x00,y00,x01,y01,x02,y02,x03,y03);
alqpar[0][0]=x00;
alqpar[1][0]=y00;

alqpar[0][1]=x01;
alqpar[1][1]=y01;

alqpar[0][2]=x02;
alqpar[1][2]=y02;

alqpar[0][3]=x03;
alqpar[1][3]=y03;
*/
if (deux_plaq()) {
    for (int i=0; i<13; i+=2) {
        fgets(ligne,sizeof(ligne),ft);
        sscanf(ligne,"%f%f%f%f%f%f\n",&x00,&x01,
        &x02,&x03,&x04,&x05);
        alqpar_2[i][0] = x00;
        alqpar_2[i][1] = x01;
        alqpar_2[i][2] = x02;
        alqpar_2[i][3] = x03;
        alqpar_2[i][4] = x04;
        alqpar_2[i][5] = x05;
        printf("%f%f%f%f%f%f\n",alqpar_2[i][0],alqpar_2[i][1],
        alqpar_2[i][2],alqpar_2[i][3],alqpar_2[i][4],alqpar_2[i][5]);
        fgets(ligne,sizeof(ligne),ft);
        sscanf(ligne,"%f%f%f%f%f%f\n",&y00,&y01,&y02,
        &y03,&y04,&y05);
    }
}

```

Sunday July 19, 2009

Align.cxx

Jul 19, 09 14:13

Align.cxx

Page 8/17

```

    alqpar_2[i+1][0] = y00;
    alqpar_2[i+1][1] = y01;
    alqpar_2[i+1][2] = y02;
    alqpar_2[i+1][3] = y03;
    alqpar_2[i+1][4] = y04;
    alqpar_2[i+1][5] = y05;
    printf("%f%f%f%f%f%f\n",alqpar_2[i+1][0],alqpar_2[i+1][1],
    alqpar_2[i+1][2],alqpar_2[i+1][3],alqpar_2[i+1][4],alqpar_2[i+1][5]);
}
fgets(ligne,sizeof(ligne),ft);
sscanf(ligne,"%f%f%f%f%f%f\n",&x00,&x01,&x02,&x03,&x04,&x05);
alqpar_2[14][0] = x00;
alqpar_2[14][1] = x01;
alqpar_2[14][2] = x02;
alqpar_2[14][3] = x03;
alqpar_2[14][4] = x04;
alqpar_2[14][5] = x05;
printf("%f%f%f%f%f%f\n",alqpar_2[14][0],alqpar_2[14][1],alqpar_2[14][2],al
qpar_2[14][3],
    alqpar_2[14][4],alqpar_2[14][5]);
}
fclose(ft);

return;
}

void Align::LireFichAmasPar(int run){

    char alqdir[255];
    char amasfich[255];
    sprintf(alqdir,"../align" );
    sprintf(amasfich,"%s/amaspar_%d.dat",alqdir,run);
    printf("amasfich %s\n",amasfich);
    FILE *ft;
    ft = fopen(amasfich,"r");
    if(!ft){
        printf("Error cannot open %s\n",amasfich);
        return;
    }
    int n00,n01,n02,n03,n04,n05;
    char ligne[100];
    for (int i=0; i<2; i++) {
        fgets(ligne,sizeof(ligne),ft);
        sscanf(ligne,"%d%d%d%d%d%d\n",&n00,&n01,
        &n02,&n03,&n04,&n05);
        amaspar[i][0] = n00;
        amaspar[i][1] = n01;
        amaspar[i][2] = n02;
        amaspar[i][3] = n03;
        amaspar[i][4] = n04;
        amaspar[i][5] = n05;
        printf("i %d: %d %d %d %d %d\n",i,amaspar[i][0],amaspar[i][1],
        amaspar[i][2],amaspar[i][3],amaspar[i][4],amaspar[i][5]);
    }
    fclose(ft);

    return;
}

void Align::LireFonctionEta(int run){

    char alqdir[255];
    char etafich[255];
    char chist[40];
    sprintf(alqdir,"../align" );
    sprintf(etafich,"%s/feta_%d.root",alqdir,run);
    printf("etafich %s\n",etafich);
    TFile *feta = new TFile(etafich);
    for (int i=0; i<6; i++) {

```

4/9

Jul 19, 09 14:13

Align.cxx

Page 9/17

```

    sprintf(chist, "feta_p%d", i);
    printf("%s\n", chist);
    feta_p[i] = (TH1D *) feta->Get(chist)->Clone();
    feta_p[i]->SetDirectory(0);
    int nbinsx = feta_p[i]->GetNbinsX();
    printf("i %d nbins p %d\n", i, nbinsx);
    sprintf(chist, "feta_n%d", i);
    printf("%s\n", chist);
    feta_n[i] = (TH1D *) feta->Get(chist)->Clone();
    feta_n[i]->SetDirectory(0);
}
feta->Close();
for (int i=0; i<6; i++) {
    int nbinsx = feta_p[i]->GetNbinsX();
    printf("i %d nbins p %d\n", i, nbinsx);
    nbinsx = feta_n[i]->GetNbinsX();
    printf("i %d nbins n %d\n", i, nbinsx);
}

return;
}

void Align::LireFonctionEta3(int run){

    char algdir[255];
    char eta3fich[255];
    char chist[40];
    sprintf(algdir, "../align" );
    sprintf(eta3fich, "%s/feta3_%d.root", algdir, run);
    printf("eta3fich %s\n", eta3fich);
    TFile *feta3 = new TFile(eta3fich);
    for (int i=0; i<6; i++)
        if (i == 3) {
            //    sprintf(chist, "feta_p%d", i);
            //    printf("%s\n", chist);
            //    feta_p[i] = (TH1D *) feta->Get(chist)->Clone();
            //    feta_p[i]->SetDirectory(0);
            //    int nbinsx = feta_p[i]->GetNbinsX();
            //    printf("i %d nbins p %d\n", i, nbinsx);
            sprintf(chist, "feta3_n%d", i);
            printf("%s\n", chist);
            feta3_n[i] = (TH1D *) feta3->Get(chist)->Clone();
            feta3_n[i]->SetDirectory(0);
        };
    feta3->Close();
    for (int i=0; i<6; i++)
        if (i == 3) {
            //    int nbinsx = feta_p[i]->GetNbinsX();
            //    printf("i %d nbins p %d\n", i, nbinsx);
            int nbinsx = feta3_n[i]->GetNbinsX();
            printf("i %d nbins n %d\n", i, nbinsx);
            for (int ib=1; ib<nbinsx+1; ib++)
                if (feta3_n[i]->GetBinContent(ib) == 0 && feta3_n[i]->GetBinCenter(ib) >
bord_eta3_n[i]) bord_eta3_n[i] = feta3_n[i]->GetBinCenter(ib);
        };
    printf("bord %f\n", bord_eta3_n[3]);

    return;
}

double Align::ligne_projx(int i, int j, int k, Trace* tra) {
    double xpos = -999;
    double m, b;
    if (tra->z[i] == tra->z[j]) return(xpos);
    m = (tra->x[i]-tra->x[j])/(tra->z[i]-tra->z[j]);
    b = tra->z[j]*tra->x[i] - tra->z[i]*tra->x[j];
    b/= (tra->z[j]-tra->z[i]);
    xpos = m*tra->z[k] + b;
    //    printf("xpos %f\n", xpos);
}

```

Jul 19, 09 14:13

Align.cxx

Page 10/17

```

    }
    return(xpos);
}

double Align::ligne_projy(int i, int j, int k, Trace* tra) {
    double ypos = -999;
    double m, b;
    if (tra->z[i] == tra->z[j]) return(ypos);
    m = (tra->y[i]-tra->y[j])/(tra->z[i]-tra->z[j]);
    b = tra->z[j]*tra->y[i] - tra->z[i]*tra->y[j];
    b/= (tra->z[j]-tra->z[i]);
    ypos = m*tra->z[k] + b;
    return(ypos);
}

double Align::ligne_projxz(int i, int j, float z, Trace* tra) {
    double xpos = -999;
    double m, b;
    //    printf("x1 %f y1 %f z1 %f x2 %f y2 %f z2 %f\n", apos(i,0),
    //        apos(i,1), apos(i,2), apos(j,0), apos(j,1), apos(j,2));
    if (tra->z[i] == tra->z[j]) return(xpos);
    m = (tra->x[i]-tra->x[j])/(tra->z[i]-tra->z[j]);
    b = tra->z[j]*tra->x[i] - tra->z[i]*tra->x[j];
    b/= (tra->z[j]-tra->z[i]);
    xpos = m*z + b;
    //    printf("neven %d i %d j %d z %f xpos %f\n", nevent, i, j, z, xpos);
    return(xpos);
}

double Align::ligne_projyz(int i, int j, float z, Trace* tra) {
    double ypos = -999;
    double m, b;
    if (tra->z[i] == tra->z[j]) return(ypos);
    m = (tra->y[i]-tra->y[j])/(tra->z[i]-tra->z[j]);
    b = tra->z[j]*tra->y[i] - tra->z[i]*tra->y[j];
    b/= (tra->z[j]-tra->z[i]);
    ypos = m*z + b;
    return(ypos);
}

double Align::div_dxdz(int i, int j, Trace* tra) {
    double dx = tra->x[i] - tra->x[j];
    //    printf("xi %f xj %f\n", tra->x[i], tra->x[j]);
    double dz = tra->z[i] - tra->z[j];
    //    printf("zi %f zj %f\n", tra->z[i], tra->z[j]);
    //    printf("dx %f dz %f\n", dx, dz);
    double val = dx/dz;
    if (tra->z[j] > tra->z[i]) val=-val;
    return(val);
}

double Align::div_dydz(int i, int j, Trace* tra) {
    double dy = tra->y[i] - tra->y[j];
    double dz = tra->z[i] - tra->z[j];
    //    printf(" ztdr1 %d z1 %f ztdr2 %d z2 %f\n", ztdr[i], tra->z[ztdr[i]], ztdr[j],
    tra->z[ztdr[j]]);
    double val = dy/dz;
    if (tra->z[j] > tra->z[i]) val=-val;
    return(val);
}

double Align::div_dxdz_lin(Trace* tra, int exclu) {
    int n = 0;
    for (int i=0; i<6; i++) {
        if (i != exclu) {
            xp[n] = tra->x[i];
            zp[n] = tra->z[i];
            //            printf(" n %d x %f z %f\n", n, xp[n], zp[n]);

```

Jul 19, 09 14:13

Align.cxx

Page 11/17

```

    e[n] = 1.;
    n++;
}
if (n < 5) return(-999.);
// printf("fit dxdz\n");
// TLinearFitter *resfitx = new TLinearFitter(1,"pol1");
// resfit->StoreData(0);
// TLinearFitter *fitxz = new TLinearFitter(1);
// TLinearFitter *fityz = new TLinearFitter(1);
fitxz->AssignData(n,1,zp,yp,e);
fitxz->Eval();
// printf("iret %d\n",iret);
// float chisqr=fitxz->GetChisquare();
// fitxz->PrintResults(3,0.);
TVectorD retpar;
fitxz->GetParameters(retpar);
// printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
if (n == 6) {
    tra->dxdz = retpar[1];
    for (int i=0; i<6; i++)
        resx[i] = xp[i] - (zp[i]*retpar[1] + retpar[0]);
    // printf(" i %d x %f z %f res %f\n",n,xp[i],zp[i],resx[i]); }
}
if (n == 5) {
    xpospred[exclu] = tra->z[exclu]*retpar[1] + retpar[0];
    resex[exclu] = tra->x[exclu] - (tra->z[exclu]*retpar[1] + retpar[0]);
}
double val=retpar[1];
fitxz->ClearPoints();
return(val);
}

```

```
void Align::div_dxdz_lin_ref(Trace* tra) {
```

```

    for (int i=0; i<4; i++) {
        int n = 0;
        xp[n] = tra->x[i];
        zp[n] = tra->z[i];
        e[n] = 1.;
        n++;
        xp[n] = tra->x[4];
        zp[n] = tra->z[4];
        e[n] = 1.;
        n++;
        xp[n] = tra->x[5];
        zp[n] = tra->z[5];
        e[n] = 1.;
        n++;
        fitxz->AssignData(n,1,zp,yp,e);
        fitxz->Eval();
        // printf("iret %d\n",iret);
        float chisqr=fitxz->GetChisquare();
        // fitxz->PrintResults(3,0.);
        TVectorD retpar;
        fitxz->GetParameters(retpar);
        // printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
        ressx[i] = tra->x[i] - (tra->z[i]*retpar[1] + retpar[0]);
        fitxz->ClearPoints();
    }
    return;
}

```

```
double Align::div_dydz_lin(Trace* tra, int exclu) {
```

```

    int n = 0;
    for (int i=0; i<6; i++) {
        if (i != exclu) {
            yp[n] = tra->y[i];

```

Jul 19, 09 14:13

Align.cxx

Page 12/17

```

        zp[n] = tra->z[i];
        // printf(" n %d x %f z %f\n",n,yp[n],zp[n]);
        e[n] = 1.;
        n++;
    }
}
if (n < 5) return(-999.);
// printf("fit dxdz\n");
// TLinearFitter *resfitx = new TLinearFitter(1,"pol1");
// resfit->StoreData(0);
// TLinearFitter *fityz = new TLinearFitter(1);
fityz->AssignData(n,1,zp,yp,e);
fityz->Eval();
// printf("iret %d\n",iret);
float chisqr=fityz->GetChisquare();
// fitxz->PrintResults(3,0.);
TVectorD retpar;
fityz->GetParameters(retpar);
// printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
if (n == 6) {
    tra->dydz = retpar[1];
    for (int i=0; i<6; i++)
        resy[i] = yp[i] - (zp[i]*retpar[1] + retpar[0]);
    // printf(" i %d x %f z %f res %f\n",n,xp[i],zp[i],resx[i]); }
}
if (n == 5) {
    ypospred[exclu] = tra->z[exclu]*retpar[1] + retpar[0];
    resy[exclu] = tra->y[exclu] - (tra->z[exclu]*retpar[1] + retpar[0]);
}
double val=retpar[1];
fityz->ClearPoints();
return(val);
}

```

```
void Align::div_dydz_lin_ref(Trace* tra) {
```

```

    for (int i=0; i<4; i++) {
        int n = 0;
        yp[n] = tra->y[i];
        zp[n] = tra->z[i];
        e[n] = 1.;
        n++;
        yp[n] = tra->y[4];
        zp[n] = tra->z[4];
        e[n] = 1.;
        n++;
        yp[n] = tra->y[5];
        zp[n] = tra->z[5];
        e[n] = 1.;
        n++;
        fityz->AssignData(n,1,zp,yp,e);
        fityz->Eval();
        // printf("iret %d\n",iret);
        float chisqr=fityz->GetChisquare();
        // fitxz->PrintResults(3,0.);
        TVectorD retpar;
        fityz->GetParameters(retpar);
        // printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
        ressy[i] = tra->y[i] - (tra->z[i]*retpar[1] + retpar[0]);
        fityz->ClearPoints();
    }
    return;
}

```

```
double Align::div_dxdz_lin_proj_pos0(Trace* tra) {
```

```

    int n = 0;
    for (int i=0; i<4; i++) {
        if (i != 2) {

```

Jul 19, 09 14:13

Align.cxx

Page 13/17

```

    xp[n] = tra->x[i];
    zp[n] = tra->z[i];
    //    printf(" n %d x %f z %f\n",n,xp[n],zp[n]);
    e[n] = 1.;
    n++;
}
//    printf("fit dxdz\n");
//    TLinearFitter *resfitx = new TLinearFitter(1,"poll");
//    resfit->StoreData(0);
//    TLinearFitter *fitxz = new TLinearFitter(1);
//    sans le petit a la position 0
fitxz->AssignData(n,l,zp,xp,e);
fitxz->Eval();
//    printf("iret %d\n",iret);
float chisqr=fitxz->GetChisquare();
//    fitxz->PrintResults(3,0.);
TVectorD retpar;
fitxz->GetParameters(retpar);
//    printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
resx_pos0[1] = tra->x[5] - (tra->z[5]*retpar[1] + retpar[0]);
//    printf(" i %d x %f z %f res %f\n",n,xp[i],zp[i],resx[i]); }
fitxz->ClearPoints();
xp[n] = tra->x[5];
zp[n] = tra->z[5];
e[n] = 1.;
n++;
// avec le petit a la position 0
fitxz->AssignData(n,l,zp,xp,e);
fitxz->Eval();
chisqr=fitxz->GetChisquare();
//    fitxz->PrintResults(3,0.);
fitxz->GetParameters(retpar);
//    printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
resx_pos0[0] = xp[3] - (zp[3]*retpar[1] + retpar[0]);
//    printf(" i %d x %f z %f res %f\n",n,xp[i],zp[i],resx[i]); }
double val=retpar[1];
fitxz->ClearPoints();
return(val);
}

double Align::div_dydz_lin_proj_pos0(Trace* tra) {

    int n = 0;
    for (int i=0; i<4; i++) {
        if (i != 2) {
            yp[n] = tra->y[i];
            zp[n] = tra->z[i];
            //    printf(" n %d y %f z %f\n",n,yp[n],zp[n]);
            e[n] = 1.;
            n++;
        }
    }
    //    printf("fit dxdz\n");
    //    TLinearFitter *resfitx = new TLinearFitter(1,"poll");
    //    resfit->StoreData(0);
    //    TLinearFitter *fityz = new TLinearFitter(1);
    //    sans le petit a la position 0
    fityz->AssignData(n,l,zp,yp,e);
    fityz->Eval();
    //    printf("iret %d\n",iret);
    float chisqr=fityz->GetChisquare();
    //    fitxz->PrintResults(3,0.);
    TVectorD retpar;
    fityz->GetParameters(retpar);
    //    printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
    resy_pos0[1] = tra->y[5] - (tra->z[5]*retpar[1] + retpar[0]);
    //    printf(" i %d x %f z %f res %f\n",n,xp[i],zp[i],resx[i]); }
    fityz->ClearPoints();
}

```

Sunday July 19, 2009

Align.cxx

Jul 19, 09 14:13

Align.cxx

Page 14/17

```

    yp[n] = tra->y[5];
    zp[n] = tra->z[5];
    e[n] = 1.;
    n++;
    // avec le petit a la position 0
    fityz->AssignData(n,l,zp,yp,e);
    fityz->Eval();
    chisqr=fityz->GetChisquare();
    //    fityz->PrintResults(3,0.);
    fityz->GetParameters(retpar);
    //    printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
    resy_pos0[0] = yp[3] - (zp[3]*retpar[1] + retpar[0]);
    //    printf(" i %d x %f z %f res %f\n",n,xp[i],zp[i],resx[i]); }
    double val=retpar[1];
    fityz->ClearPoints();
    return(val);
}

double Align::div_dxdz_lin_proj_pos4(Trace* tra) {

    int n = 0;
    for (int i=0; i<4; i++) {
        xp[n] = tra->x[i];
        zp[n] = tra->z[i];
        //    printf(" n %d x %f z %f\n",n,xp[n],zp[n]);
        e[n] = 1.;
        n++;
    }
    //    printf("fit dxdz\n");
    //    TLinearFitter *resfitx = new TLinearFitter(1,"poll");
    //    resfit->StoreData(0);
    //    TLinearFitter *fitxz = new TLinearFitter(1);
    //    sans le petit a la position 4
    fitxz->AssignData(n,l,zp,xp,e);
    fitxz->Eval();
    //    printf("iret %d\n",iret);
    float chisqr=fitxz->GetChisquare();
    //    fitxz->PrintResults(3,0.);
    TVectorD retpar;
    fitxz->GetParameters(retpar);
    //    printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
    resx_pos4[1] = tra->x[4] - (tra->z[4]*retpar[1] + retpar[0]);
    //    printf(" i %d x %f z %f res %f\n",n,xp[i],zp[i],resx[i]); }
    fitxz->ClearPoints();
    xp[n] = tra->x[4];
    zp[n] = tra->z[4];
    e[n] = 1.;
    n++;
    // avec le petit a la position 4
    fitxz->AssignData(n,l,zp,xp,e);
    fitxz->Eval();
    chisqr=fitxz->GetChisquare();
    //    fitxz->PrintResults(3,0.);
    fitxz->GetParameters(retpar);
    //    printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
    resx_pos4[0] = xp[4] - (zp[4]*retpar[1] + retpar[0]);
    //    printf(" i %d x %f z %f res %f\n",n,xp[i],zp[i],resx[i]); }
    double val=retpar[1];
    fitxz->ClearPoints();
    return(val);
}

double Align::div_dydz_lin_proj_pos4(Trace* tra) {

    int n = 0;
    for (int i=0; i<4; i++) {
        yp[n] = tra->y[i];
        zp[n] = tra->z[i];
        //    printf(" n %d y %f z %f\n",n,yp[n],zp[n]);
    }
}

```

7/9

Jul 19, 09 14:13

Align.cxx

Page 15/17

```

    e[n] = 1.;
    n++;
}
// printf("fit dxdz\n");
// TLinearFitter *resfitx = new TLinearFitter(1,"poll");
// resfit->StoreData(0);
// TLinearFitter *fityz = new TLinearFitter(1);
// sans le petit a la position 4
fityz->AssignData(n,1,zp,yp,e);
fityz->Eval();
// printf("iret %d\n",iret);
float chisqr=fityz->GetChisquare();
// fitxz->PrintResults(3,0.);
TVectorD retpar;
fityz->GetParameters(retpar);
// printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
resy_pos4[1] = tra->y[4] - (tra->z[4]*retpar[1] + retpar[0]);
// printf(" i %d x %f z %f res %f\n",n,yp[i],zp[i],resx[i]); }
fityz->ClearPoints();
yp[n] = tra->y[4];
zp[n] = tra->z[4];
e[n] = 1.;
n++;
// avec le petit a la position 4
fityz->AssignData(n,1,zp,yp,e);
fityz->Eval();
chisqr=fityz->GetChisquare();
// fityz->PrintResults(3,0.);
fityz->GetParameters(retpar);
// printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
resy_pos4[0] = yp[4] - (zp[4]*retpar[1] + retpar[0]);
// printf(" i %d x %f z %f res %f\n",n,yp[i],zp[i],resx[i]); }
double val=retpar[1];
fityz->ClearPoints();
return(val);
}

double Align::div_dxdz_lin_pos123(Trace* tra, int exclu) {
    int n = 0;
    for (int i=0; i<4; i++) {
        if (i != exclu && i != 2) {
            xp[n] = tra->x[i];
            zp[n] = tra->z[i];
            // printf(" n %d x %f z %f\n",n,xp[n],zp[n]);
            e[n] = 1.;
            n++;
        }
    }
    if (n < 2) return(-999.);
    // printf("fit dxdz\n");
    // TLinearFitter *resfitx = new TLinearFitter(1,"poll");
    // resfit->StoreData(0);
    // TLinearFitter *fitxz = new TLinearFitter(1);
    fitxz->AssignData(n,1,zp,xp,e);
    fitxz->Eval();
    // printf("iret %d\n",iret);
    float chisqr=fitxz->GetChisquare();
    // fitxz->PrintResults(3,0.);
    TVectorD retpar;
    fitxz->GetParameters(retpar);
    // printf("n %d chisqr %f parametres %f %f\n",n,chisqr,retpar[0],retpar[1]);
    ;
    if (n == 3) {
        for (int i=0; i<4; i++)
            if (i != 2)
                resxech[i] = tra->x[i] - (tra->z[i]*retpar[1] + retpar[0]);
        // printf(" i %d x %f z %f res %f\n",n,xp[i],zp[i],resx[i]); }
    }
}

```

Jul 19, 09 14:13

Align.cxx

Page 16/17

```

    if (n == 2) {
        resxech[exclu] = tra->x[exclu] - (tra->z[exclu]*retpar[1] + retpar[0]);
    }
    double val=retpar[1];
    fitxz->ClearPoints();
    return(val);
}

double Align::div_dydz_lin_pos123(Trace* tra, int exclu) {
    int n = 0;
    for (int i=0; i<4; i++) {
        if (i != exclu && i != 2) {
            yp[n] = tra->y[i];
            zp[n] = tra->z[i];
            // printf(" n %d x %f z %f\n",n,yp[n],zp[n]);
            e[n] = 1.;
            n++;
        }
    }
    if (n < 2) return(-999.);
    // printf("fit dxdz\n");
    // TLinearFitter *resfitx = new TLinearFitter(1,"poll");
    // resfit->StoreData(0);
    // TLinearFitter *fityz = new TLinearFitter(1);
    fityz->AssignData(n,1,zp,yp,e);
    fityz->Eval();
    // printf("iret %d\n",iret);
    float chisqr=fityz->GetChisquare();
    // fitxz->PrintResults(3,0.);
    TVectorD retpar;
    fityz->GetParameters(retpar);
    // printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
    if (n == 3) {
        for (int i=0; i<4; i++)
            if (i != 2) resyech[i] = tra->y[i] - (tra->z[i]*retpar[1] + retpar[0]);
        // printf(" i %d x %f z %f res %f\n",n,xp[i],zp[i],resx[i]); }
    }
    if (n == 2) {
        resyech[exclu] = tra->y[exclu] - (tra->z[exclu]*retpar[1] + retpar[0]);
    }
    double val=retpar[1];
    fityz->ClearPoints();
    return(val);
}

int Align::div_dxdz_lin_eff(Trace* tra, int exclu) {
    int n = 0;
    for (int i=0; i<6; i++) {
        if (i != exclu && tra->x[i] != -999.) {
            xp[n] = tra->x[i];
            zp[n] = tra->z[i];
            // printf(" n %d x %f z %f\n",n,xp[n],zp[n]);
            e[n] = 1.;
            n++;
        }
    }
    if (n < 5) {
        reseffx[exclu] = 999.;
        return(0);
    }
    fitxz->AssignData(n,1,zp,xp,e);
    fitxz->Eval();
    // printf("iret %d\n",iret);
    // float chisqr=fitxz->GetChisquare();
    // fitxz->PrintResults(3,0.);
    TVectorD retpar;
    fitxz->GetParameters(retpar);
}

```


Jul 19, 09 14:13

Align.cxx

Page 17/17

```

// printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
if (tra->x[exclu] != -999.)
    reseffx[exclu] = tra->x[exclu] - (tra->z[exclu]*retpar[1] + retpar[0]);
else
    reseffx[exclu] = -999.;
// printf("exclu %d reseffx %f\n",exclu,reseffx[exclu]);
double val=retpar[1];
fitxz->ClearPoints();
return(1);
}

int Align::div_dydz_lin_eff(Trace* tra, int exclu) {
    int n = 0;
    for (int i=0; i<6; i++) {
        if (i != exclu && tra->y[i] != -999.) {
            yp[n] = tra->y[i];
            zp[n] = tra->z[i];
            // printf(" n %d x %f z %f\n",n,yp[n],zp[n]);
            e[n] = 1.;
            n++;
        }
    }
    if (n < 5) {
        reseffy[exclu] = 999.;
        return(0);
    }
    fityz->AssignData(n,1,zp,yp,e);
    fityz->Eval();
    float chisqr=fityz->GetChisquare();
    // fityz->PrintResults(3,0.);
    TVectorD retpar;
    fityz->GetParameters(retpar);
    // printf("chisqr %f parametres %f %f\n",chisqr,retpar[0],retpar[1]);
    if (tra->y[exclu] != -999.)
        reseffy[exclu] = tra->y[exclu] - (tra->z[exclu]*retpar[1] + retpar[0]);
    else
        reseffy[exclu] = -999.;
    double val=retpar[1];
    fityz->ClearPoints();
    return(1);
}

```