FRIB separator group (MH,MP) requests due to the next problems:

1. Increasing of a spatial spot in some case for Distribution method calculations
2. Increasing of angular distribution widths in some case for Distribution method calculations
3. "Issues" with rotation blocks.

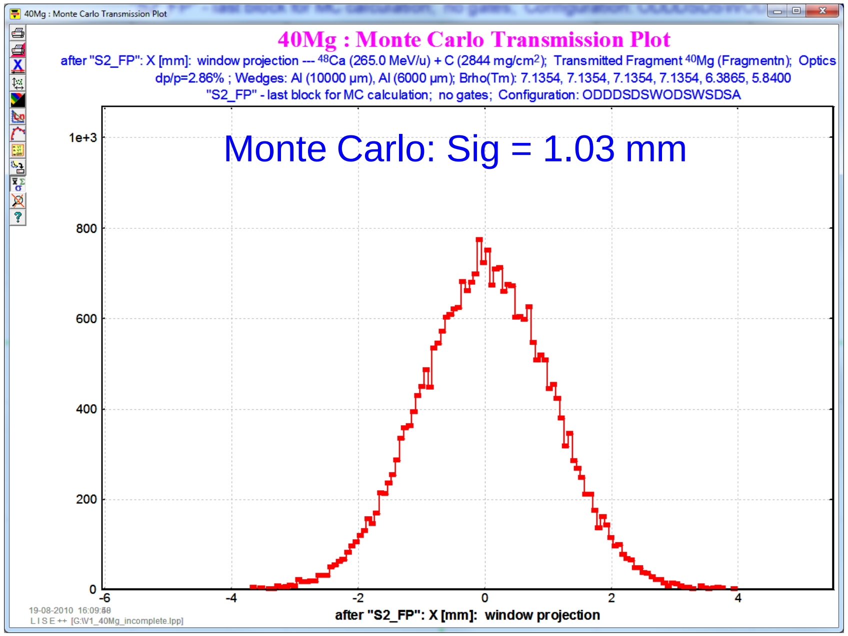
new beta-version is available via the LISE-ftp site (subdirectory "beta").

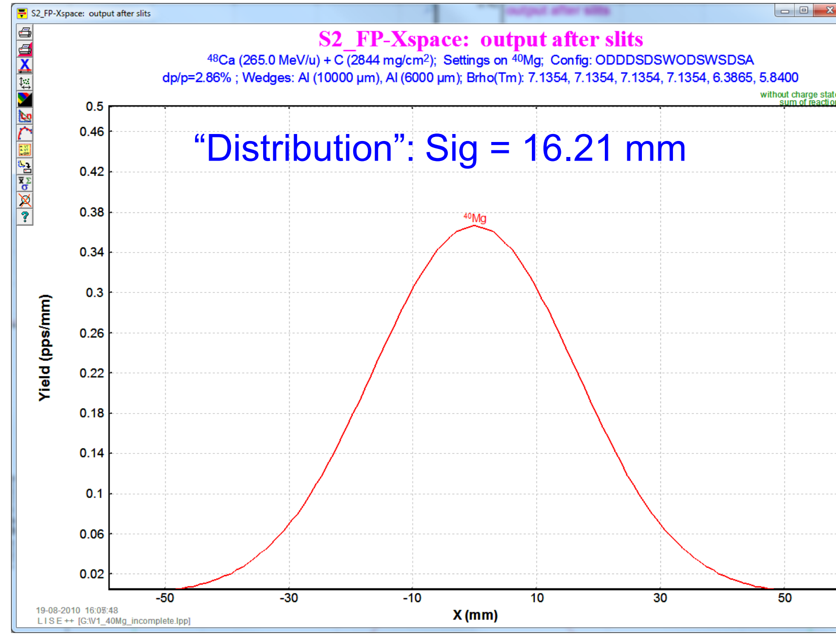
This version will be installed in directory "Program files ***\LISEbeta"
Group "LISEbeta" in folders, as well as MyDocuments\LISEbeta.
It means there is not overcrossing with the official LISE-version.

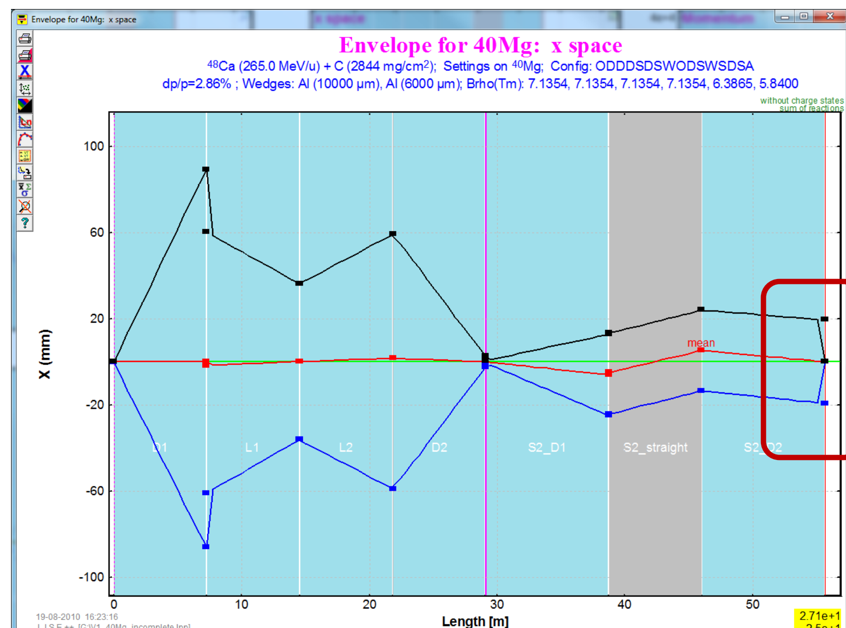
1. Increasing of a spatial spot in some case for Distribution method calculations

Example for version 9.1.13

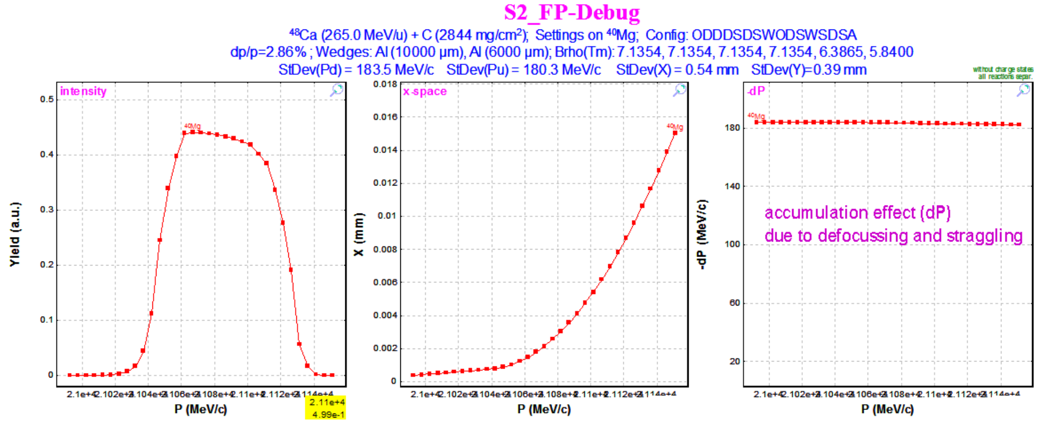
http://groups.nscl.msu.edu/lise/9_2/9_2_7/V1_40Mg_incomplete.lpp





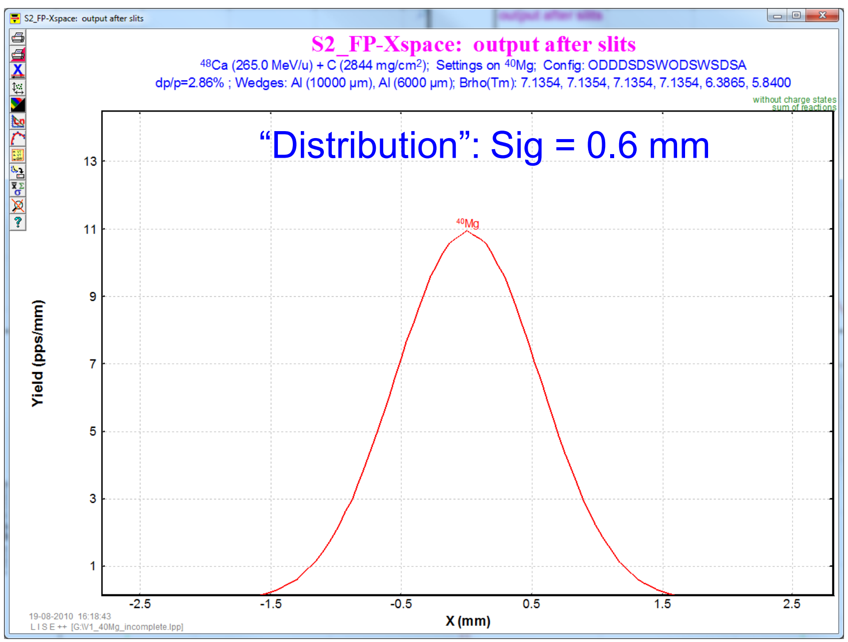


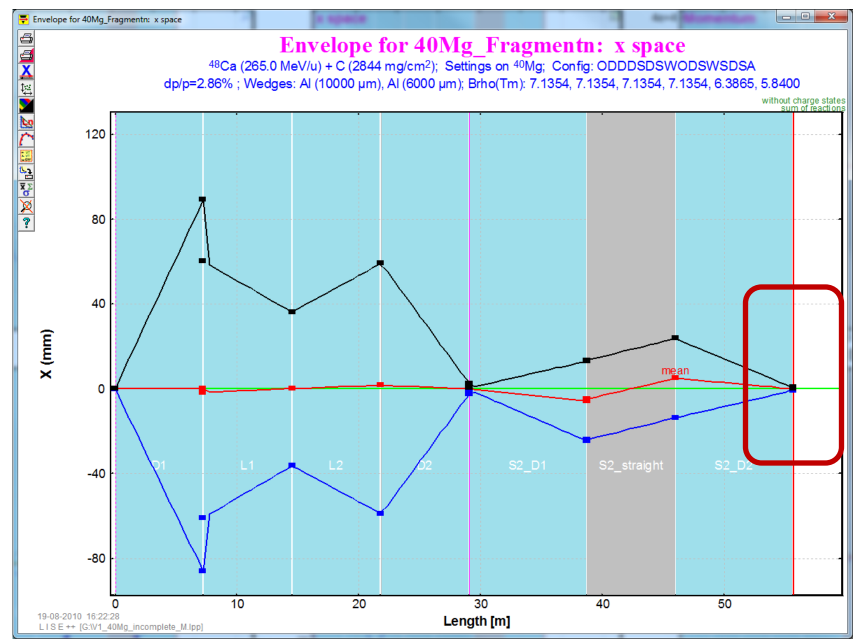
1. Increasing of a spatial spot in some case for Distribution method calculations (2)



Local dispersion coefficient has been used to produce this dPX component for the convolution.
It is important in the case of large straggling and defocusing values.
It has been corrected in the new official version 9.1.19

Example for version 9.1.19





http://groups.nscl.msu.edu/lise/9_2/9_2_7/V1_40Mg_incomplete.lpp

Remind:
Wedge "Curiosity" v.9.0.23
http://groups.nscl.msu.edu/lise/9_0/9_0_23_WedgeCuriosity.pdf

Class "distribution"

PHYSICAL REVIEW E VOLUME 50, NUMBER 5 NOVEMBER 1994

Transport integral: A method to calculate the time evolution of phase-space distributions

D. Bazin* and B. M. Sherrill
National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, Michigan 48824
(Received 7 February 1994; revised manuscript received 22 August 1994)

Each integral is now independent and corresponds to a convolution product

$$P_i'(q_i') = \frac{1}{\prod_{k=1}^{n} R_{ik}} \int_2 \cdots \int_{n-1} \bar{P}_1(q_i'-t_2)\bar{P}_2(t_2-t_3)\cdots \times[\bar{P}_{n-1}\otimes\bar{P}_n](t_{n-1}) \times dt_2\cdots dt_{n-1} .$$

Finally, the result is given by the convolution product of all $\bar{P}_j$ functions

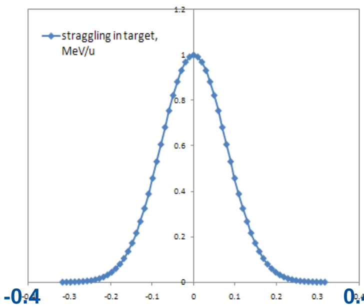
$$P_i'(q_i') = \frac{1}{\prod_{k=1}^{n} R_{ik}} [\bar{P}_1\otimes\bar{P}_2\otimes\cdots\otimes\bar{P}_n](q_i') . \quad (10)$$

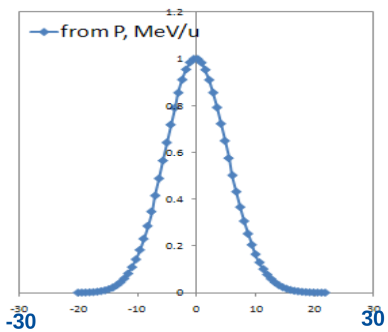
Where P_k is $I_k(x)$ (intensity distribution),
where $x_i = x_0 + i*h$,
h is the step,
$0 \le i \le N$ (distribution dimension)

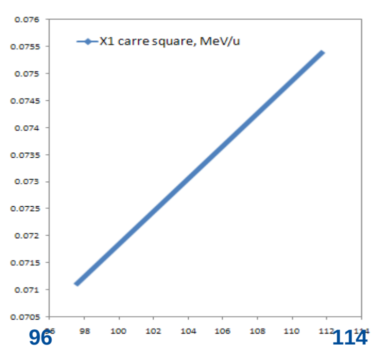
For example the energy distribution after the target:

Input:

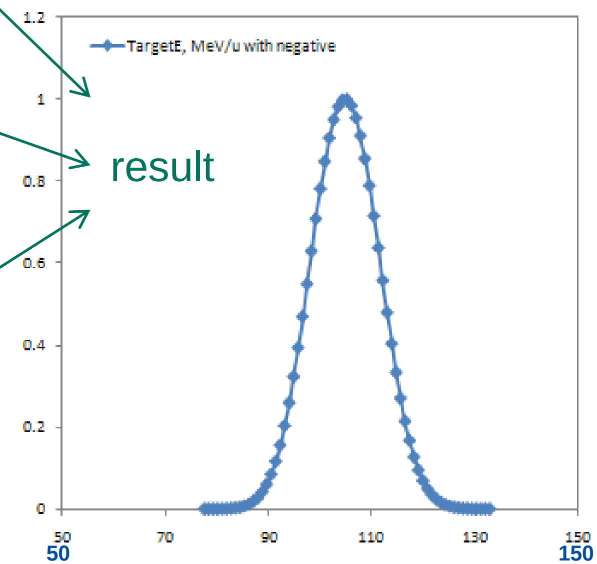
1. Beam emittance,
2. Energy straggling in target
3. Momentum distribution after reaction
4. Energy distribution due to energy loss difference in target between beam and fragment







result



Remind:
Wedge "Curiosity" v.9.0.23
http://groups.nscl.msu.edu/lise/9_0/9_0_23_WedgeCuriosity.pdf

Transformation of distributions

MICHIGAN STATE UNIVERSITY LISE++

http://groups.nscl.msu.edu/lise/4_5/lise_4_5.htm

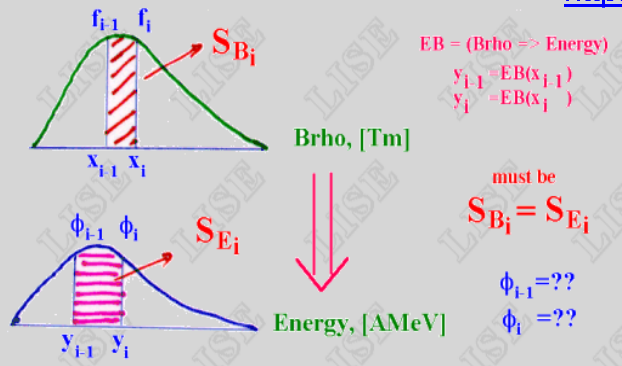


Fig.4. The scheme of conversion of one distribution in another

In a basis of conversion of one distribution in another (the scheme represented in Fig.4) lays saving of squares between every each *i-1* and *i* points. In the last versions the given task was solved rather simple way that had an effect for quality of conversions at such small dimension of distributions (NP=128).

The edge effects were especially appeared in distributions of energy, ranges in matter as they may not be negative. We shall assume that the nucleus with the certain distribution passes through substance and the *i*-point of distribution stops in matter, and following passes. Then function appropriate between *i-1* and *i* pointspoints for preservation of the area should aspires to infinity. Rather complex mechanism of smoothing was applied. But all the same it is ideal to solve this problem it was not possible!

In the last versions the area between points was determined by the next primitive expression:

$$S_i = \frac{(f_{i-1} + f_i)}{2} \cdot |x_i - x_{i-1}|$$

We may use now correct calculation of area is next:

$$S_i = \int_{x_{i-1}}^{x_i} f(x)\,dx \quad ,$$

because we have infinite function f(x) due to introduction of procedure cubic spline.
The condition of equality of the areas in both distributions between in an interval can be presented in the following kind:

$$S_{E_i} = S_{E_i} \quad \Rightarrow \quad \int_{x_{i-1}}^{x_i} f(x)\,dx = \int_{y_{i-1}}^{y_i} \phi(y)\,dy \quad .$$

Doing substitution $y=EB(x)$ it possible to get simple and good solution with application of the first derivative of the function $f(x)$:

$$\phi[y(x)] = f(x) \cdot \left(\frac{dEB}{dx}\right)^{-1} .$$

This derivative can be taken with the help of cubic spline procedure having constructed distribution *x* from *y*. Using further cubic spline procedure for f (*y*) distribution can be proceeded from complicated distribution with a variable step between points to more simple with a constant step accordingly.

Remind:
Wedge "Curiosity" v.9.0.23
http://groups.nscl.msu.edu/lise/9_0/9_0_23_WedgeCuriosity.pdf

Class "distribution2"

MICHIGAN STATE UNIVERSITY LISE++

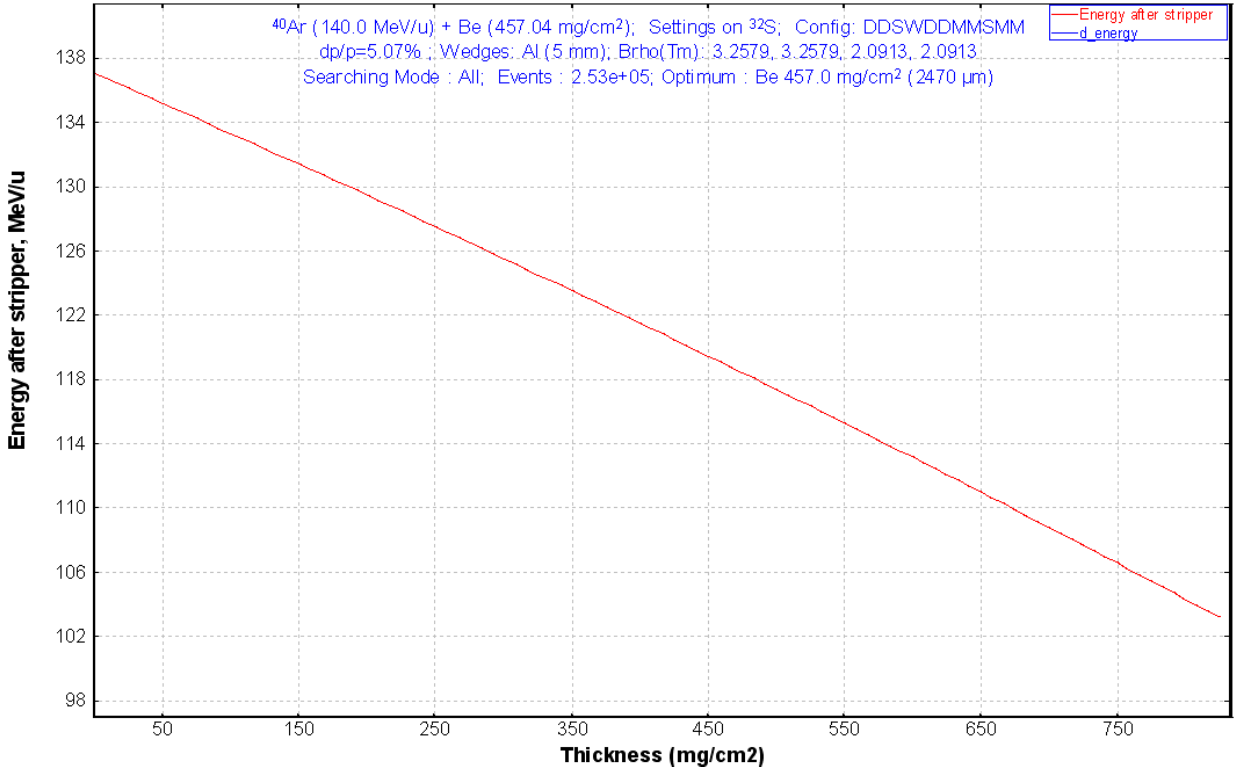
Version 4.

If the "distribution" class has just one array double I[N],

whereas the "distribution2" class
double y[N]
double x[N],

What allows easily create $x=f^{-1}(y)$ from $y=f(x)$

```
//==========================================
class distribution2 : public distribution {

public:

distribution2(double a, double b, int n ,char
*un, char *dim);
distribution2(distribution&);
distribution2(distribution2&);
~distribution2();

void operator = (distribution&);
void operator = (distribution2&);
……
```



Remind:
Wedge "Curiosity" v.9.0.23
http://groups.nscl.msu.edu/lise/9_0/9_0_23_WedgeCuriosity.pdf

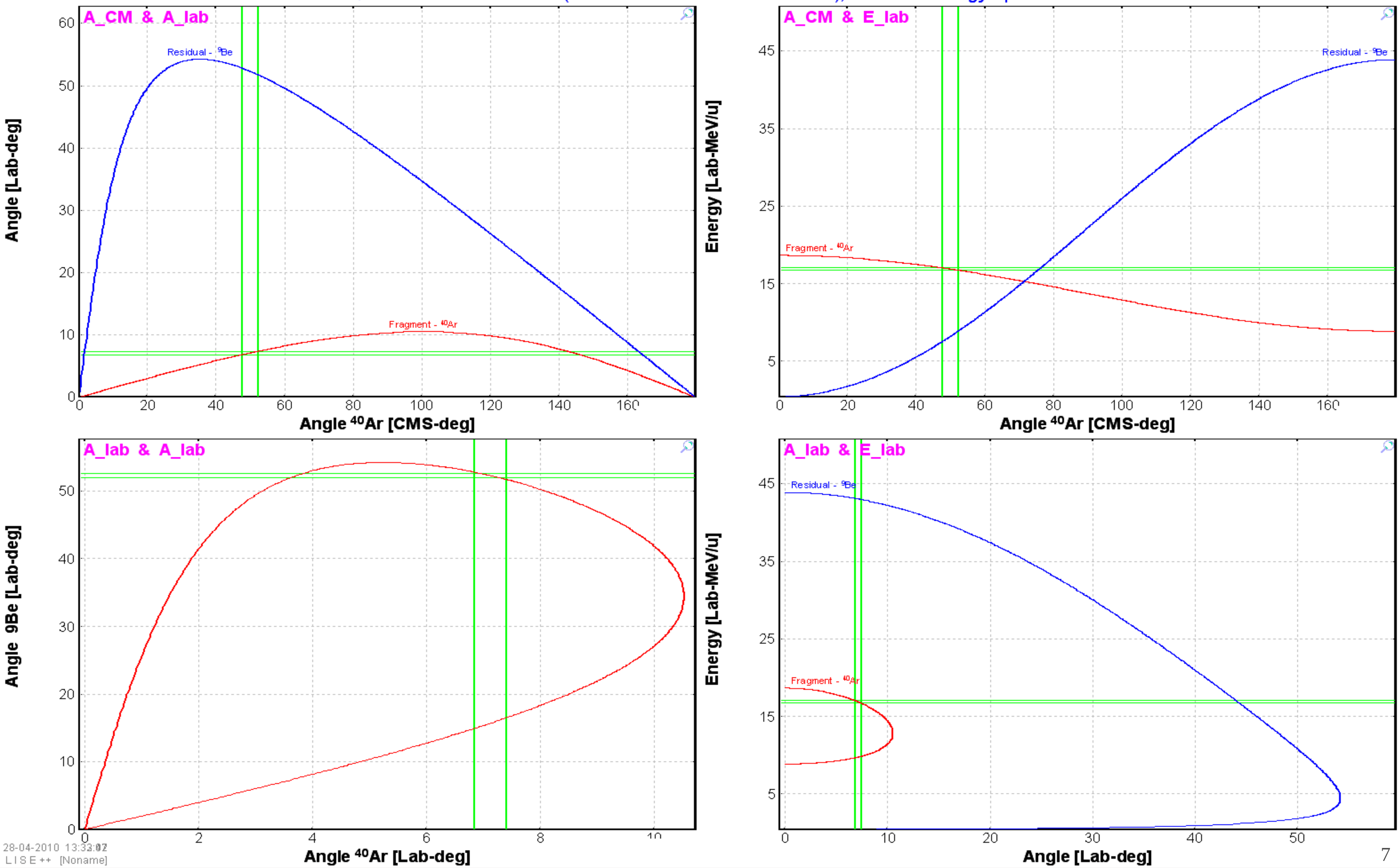
Class "distribution2" : example

Reaction's Kinematics

$^{40}Ar + {}^{9}Be \Rightarrow {}^{40}Ar + {}^{9}Be$ $^{9}Be(^{40}Ar,^{40}Ar)^{9}Be$; Reaction at the "middle" of the target

Projectile Energy at the reaction place: 20.00 MeV/u   Grazing angle in CMS [$^{40}Ar + {}^{9}Be$] = 4.63 deg

Q reaction : -50.00 MeV (Excitations 0.0+0.0=>50.0+0.0); Plotted Energy option is "after reaction"



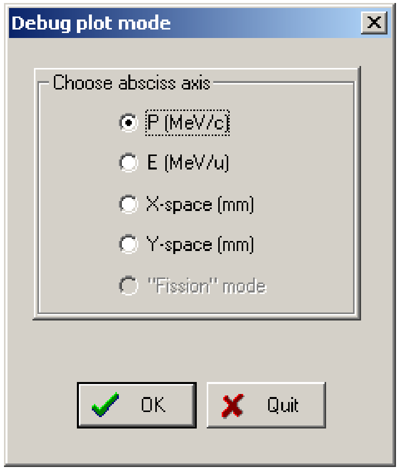
Remind:
Wedge "Curiosity" v.9.0.23
http://groups.nscl.msu.edu/lise/9_0/9_0_23_WedgeCuriosity.pdf

Class "distribution4"

MICHIGAN STATE UNIVERSITY
LISE++

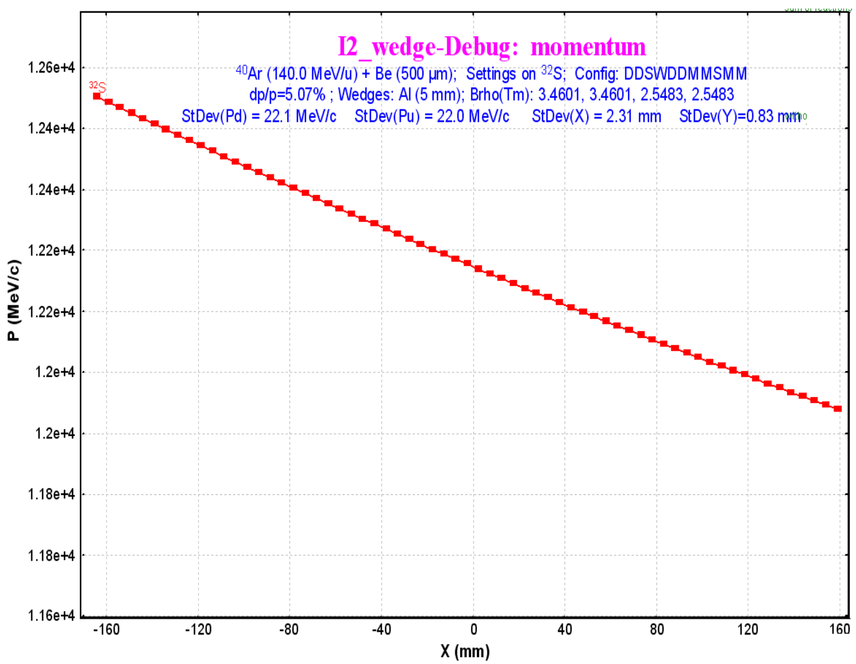
Version 6.

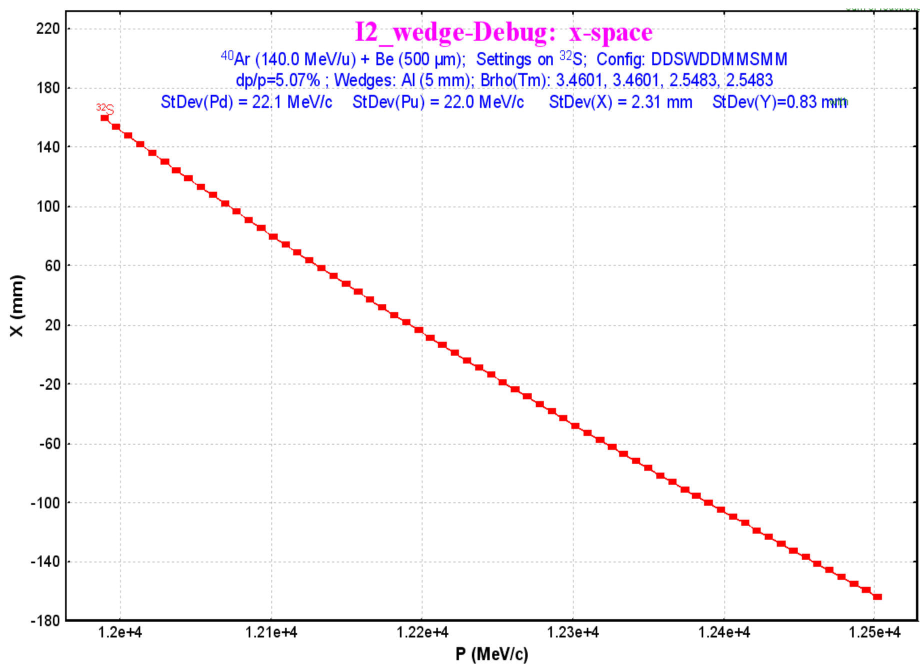
```
class distrFour{

public:
   distrFour(int Ninit=Ndistr4_XY, int mode_init=em_XY);
   distrFour(distrFour&);
  ~distrFour();

   distribution2 **d4;

  void ChangeBase(int BaseNew, bool MakeUniformOpt=false);
……..
```




```
enum edistrFour {
     e4I,
     e4P,
     e4E,
     e4X,
     e4Y,
     e4Pd,
     e4Pu,
     e4Ed,
     e4Eu
};
```

Can be "Base" (e4P, e4E, e4X, e4Y)



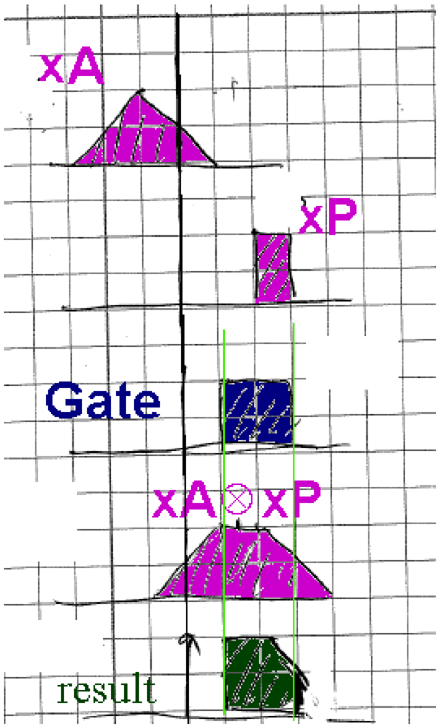


Probably X' and Y' should be included in DistrFour and be used as "Base" to solve MH effect

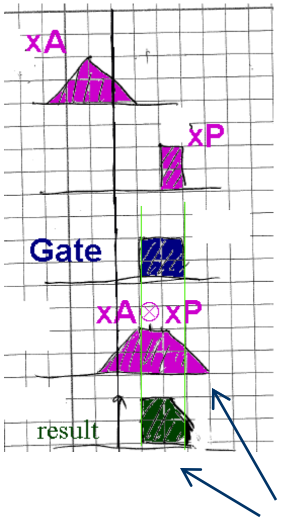
Passing spatial slits

```
//------------------------------ X-object -----------------------X->X-----------
distribution  thisX(*bcc->currS);
thisX.product(Gss);                      //  G – global coefficient

//------------------------------ X-object   (inverse Y->X)-----------------------
distribution  thisX_i(*bcc_i->currS);
thisX_i.product(Gss_i);

//------------------------------ T->X ---------------
distribution  thisA(*bcc->currA);
thisA.product(Gsa);

//------------------------------ P->X ---------------
distribution  thisA_i(*bcc_i->currA);
thisA_i.product(Gsa_i);

//------------------------------ momentum -----dp/p - > X   ----------
distribution  thisP ;  //  lately
***

//------------------------------ accumulated straggling---------------
distribution  thisPX;
****


d[0]=&thisX;
d[1]=&thisA;
d[2]=&thisX_i;
d[3]=&thisA_i;
d[4]=&this_dPX;
d[5]=&thisP;
```

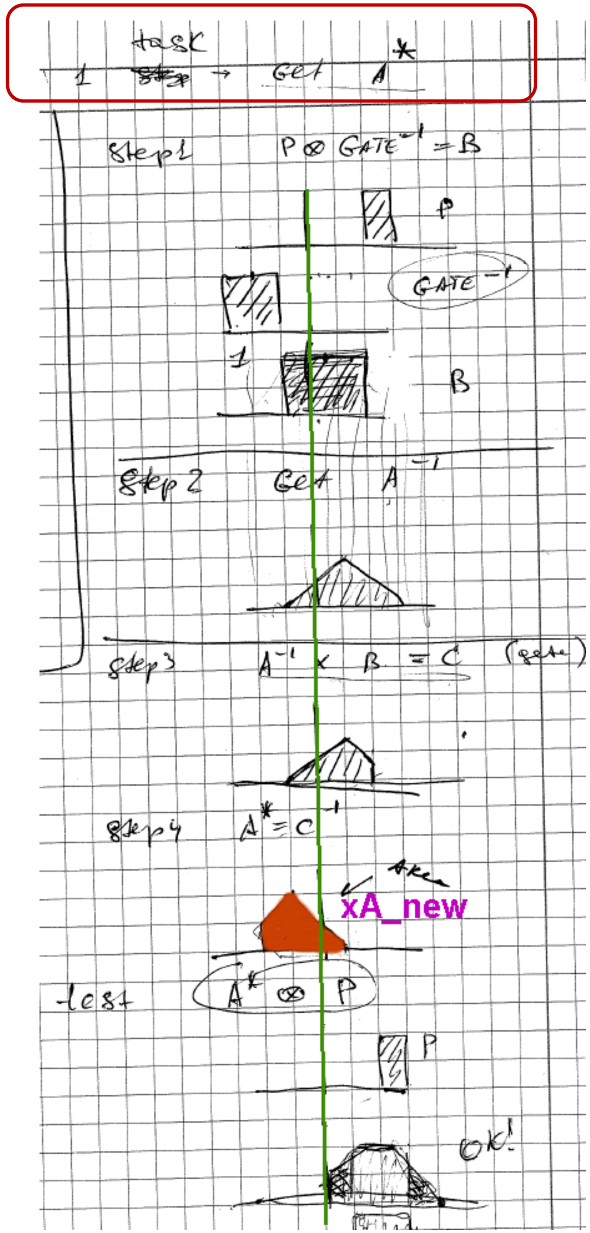
```
bc->oU_disS->convolute(6,d);


// transmission through slits

if(bc->slit.GetWork())
                bc->calc_s =  bc->oU_disS->relation(
                                  bc->slit.M(),
                                  bc->slit.P(),
                                   false, true);
```

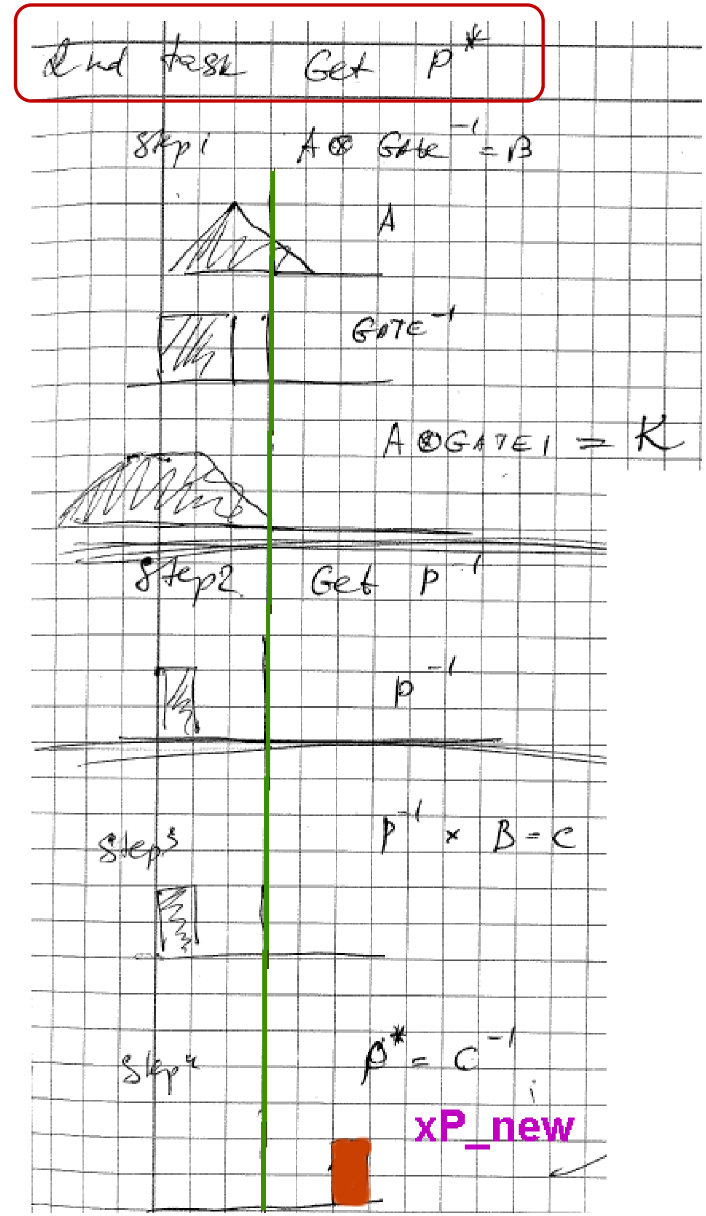


Cut of Initial distributions after passing slits



Area ratio (green/pink) is transmission





xA_new ?
A_new?

xP_new?
P_new?

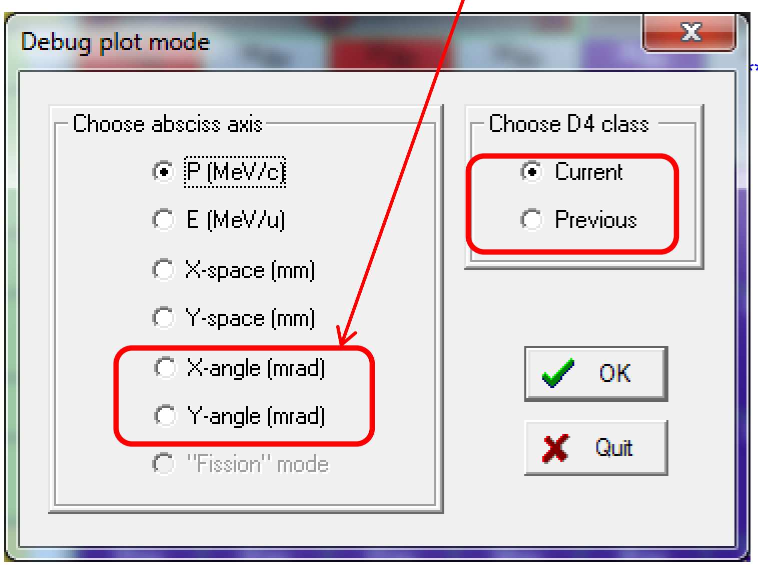
Passing spatial slits

```
enum edistrFour {
    e4I,
    e4P,
    e4E,
    e4X,
    e4Y,
    e4AX,
    e4AY,
    e4Pd,
    e4Pu,
    e4Ed,
    e4Eu
};
```

Can be "Base" (e4P, e4E, e4X, e4Y, e4AX, e4AY)

New in 9.2.7

```
distribution2 *d4P =  DF4->d4[e4P];    // P(P)
distribution2 *d4X =  DF4->d4[base];   // X(P)
distribution2 *d4A =  DF4->d4[baseA];  // A(P)
distribution2 *d4X_i =  DF4->d4[base_i];   // X'(P)
distribution2 *d4A_i =  DF4->d4[baseA_i];  // A'(P)

for( i=0;i<=NP;i++)
         {
         xP   = (d4P->ff(i)  - TauliseMomentumBlockSet0[direct]) * Lsd_coef;
         xx   =  d4X->ff(i) * Lss_coef;
         xa   =  d4A->ff(i) * Lsa_coef;
         xx_i =  d4X_i->ff(i) * Lss_i;
         xa_i =  d4A_i->ff(i) * Lsa_i;

         if(KickerUse)                                   // Kicker stuff
                 xk = d4P->ff(i)*Kicker->Taulise_kP + Kicker->Taulise_bP;

         axis_X[i]=xP+xk+xx+xa+xx_i+xa_i;                // in mm
         }
****

         int Curios = Wedge_Curiosity_Analysis(d4P,
                                    axis_X, axis_X_new, i_Imax);

         DF4->ChangeBase(base,true);

//---------------------------------------------------

         thisP = *DF4->d4[e4I]; // to transform distribution2 -> distribution1
         thisP2=  DF4->d4[e4I];
         thisP.normalization();
```



Class "Distribution4" in v.9.2.7

Passing angular acceptance

```
//--------------------------------ANGLE-----      T->T
distribution thisA(*bcc->currA);
double g_aa = B->gom->o_aa(direct);
thisA.product(g_aa);

//--------------------------------ANGLE-----      P->T
distribution thisA_i(*bcc_i->currA);
double g_aai = (direct == edirect_X ? B->gom->o_tp : B->gom->o_pt);
thisA_i.product(g_aai);

//------------------------------X-OBJECT-------- X->T
distribution thisX(*bcc->currS);
double g_ax = B->gom->o_as(direct);
thisX.product(g_ax);

//------------------------------XOBJECT-------- Y->T
distribution thisX_i(*bcc_i->currS);
double g_axi = (direct == edirect_X ? B->gom->o_ty : B->gom->o_px);

thisX_i.product(g_axi);

//------------------------------MOMENT----------- P->T
****

//----------------------------------------  Result convolution
                    d[0]=&thisX;
                    d[1]=&thisX_i;
                    d[2]=&thisP;
                    d[3]=&thisA_i;
                    d[4]=&thisA;
                    bc->o_disA->convolute(5,d,2, FlagLogTrue);
```

Momentum : D4-distribution

```
distribution2 *d4Pprev =  DF4prev->d4[e4P];          // P(P)
distribution2 *d4Aprev =  DF4prev->d4[base];          // A(P)
distribution2 *d4Xprev =  DF4prev->d4[baseX];         // X(P)
distribution2 *d4Aprev_i =  DF4prev->d4[base_i];      // Y'(P)
distribution2 *d4Xprev_i =  DF4prev->d4[baseX_i];     // X'(P)

for(int  i=0;i<=NP;i++)
 {
 aP =  (d4Pprev->ff(i)  - MomentumBlockSet0[direct]) * Lad_coef;
 aa =   d4Aprev->ff(i) * Laa_coef;
 ax =   d4Xprev->ff(i) * Las_coef;
 aa_i =   d4Aprev_i->ff(i) * Laa_i;
 ax_i =   d4Xprev_i->ff(i) * Las_i;

  axis_A[i] = aP + aa + ax + aa_i + ax_i;           // in mrad
  }

  //--------------------------------------------------

int   i_Imax = DF4prev->d4[e4I]->max_i();
Wedge_Curiosity_Analysis(d4Pprev, axis_A, axis_A_new, i_Imax);

for(int i=0; i<=NP; i++)
                     d4A_temp->input_fn(i, axis_A_new[i]);
```

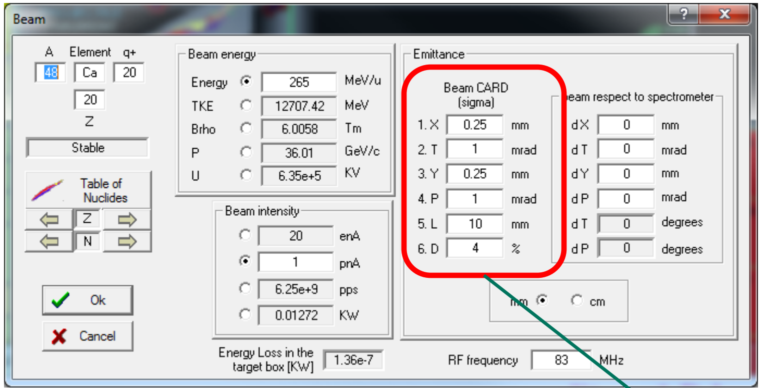
new

+ fission case….. ☹

Cut of distributions after passing angular acceptance

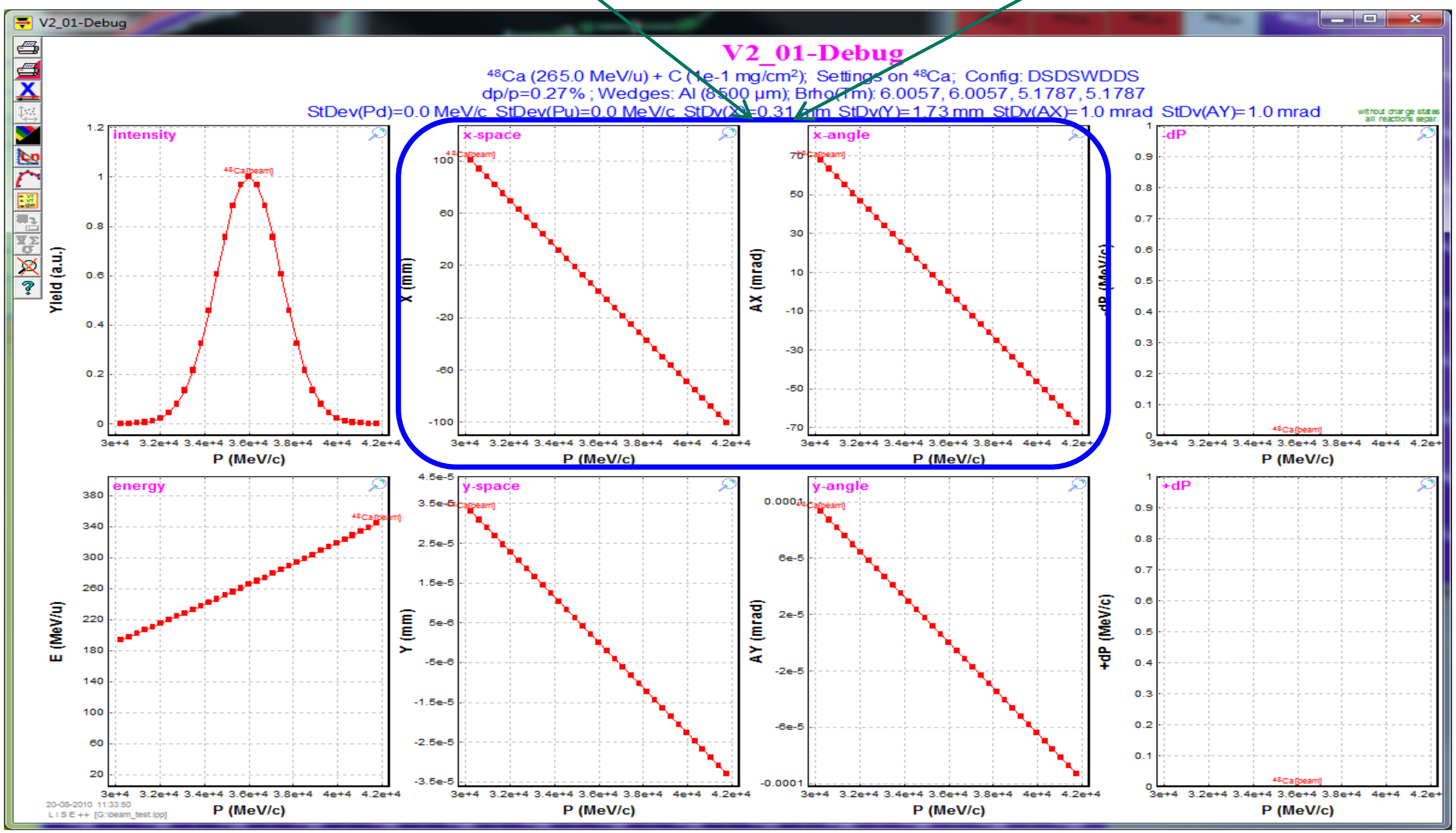
Primary beam case with large energy emittance.

http://groups.nscl.msu.edu/lise/9_2/9_2_7/beam_test.lpp



Block matrix

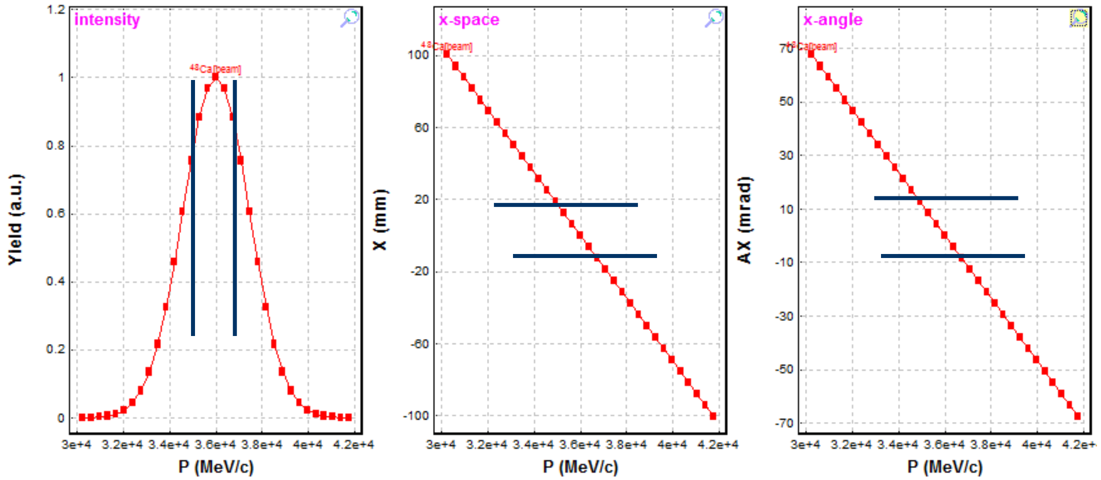
1. X	-1.24629	-0.0002	0	0	0	-6.2798
2. T	-0.20291	-0.80241	0	0	0	-4.2237
3. Y	0	0	-1.89376	1.6842	0	0
4. F	0	0	-0.51779	-0.06757	0	0
5. L	-0.39897	0.5038	0	0	1	-0.5674
6. D	0	0	0	0	0	1



Cut of distributions after passing angular acceptance

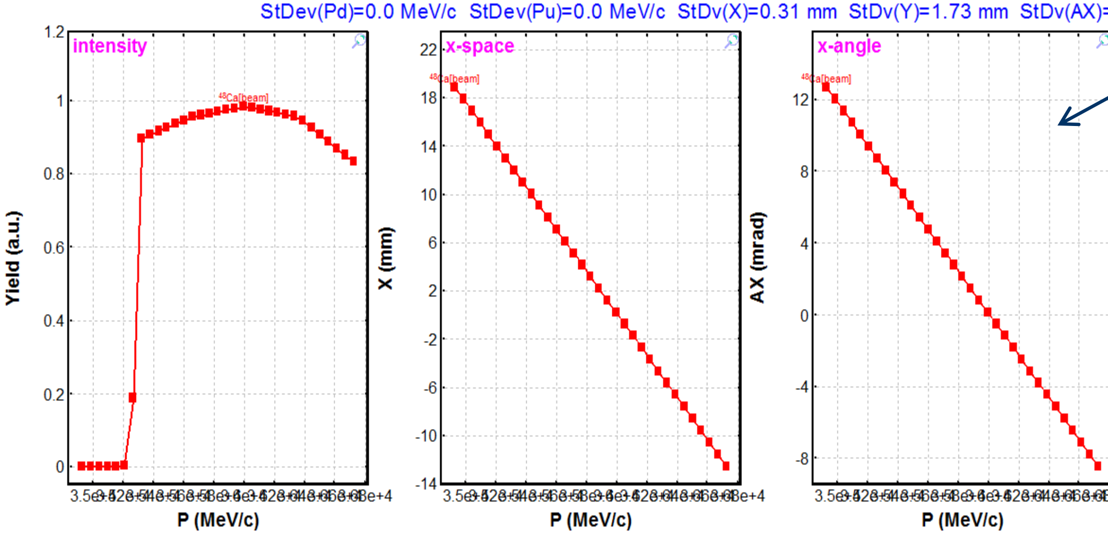
V2_01-Debug

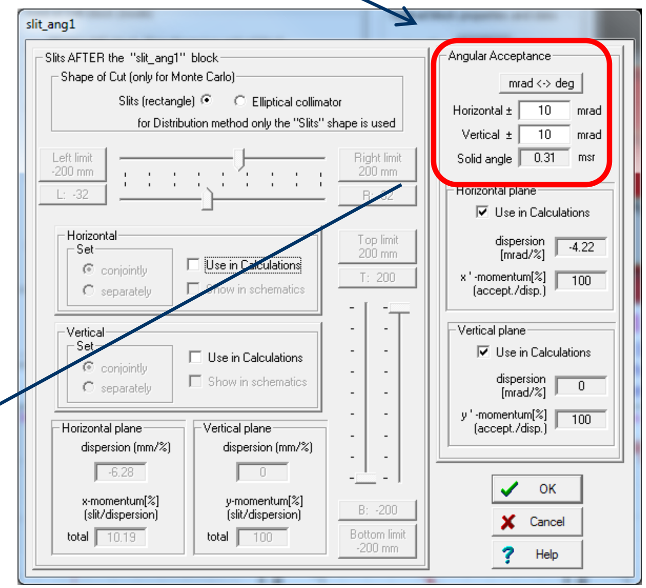
^{48}Ca (265.0 MeV/u) + C (1e-1 mg/cm^2); Settings on ^{48}Ca; Config: DSDSW
dp/p=0.27% ; Wedges: Al (8500 μm); Brho(Tm): 6.0057, 6.0057, 5.1787, 5
StDev(Pd)=0.0 MeV/c StDev(Pu)=0.0 MeV/c StDv(X)=0.31 mm StDv(Y)=1.73 mm StDv(AX)=1.



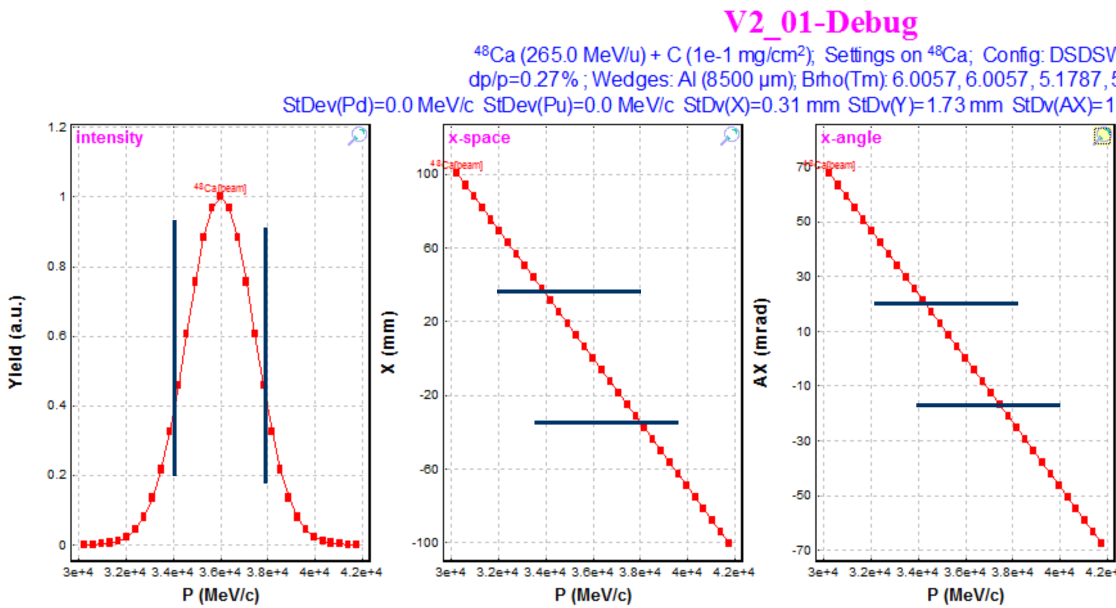
slit_ang1-Debug

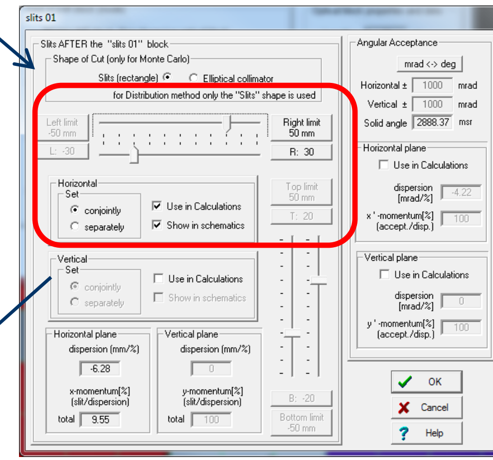
^{48}Ca (265.0 MeV/u) + C (1e-1 mg/cm^2); Settings on ^{48}Ca; Config: DSDS
dp/p=0.27% ; Wedges: Al (8500 μm); Brho(Tm): 6.0057, 6.0057, 5.1787,
StDev(Pd)=0.0 MeV/c StDev(Pu)=0.0 MeV/c StDv(X)=0.31 mm StDv(Y)=1.73 mm StDv(AX)=

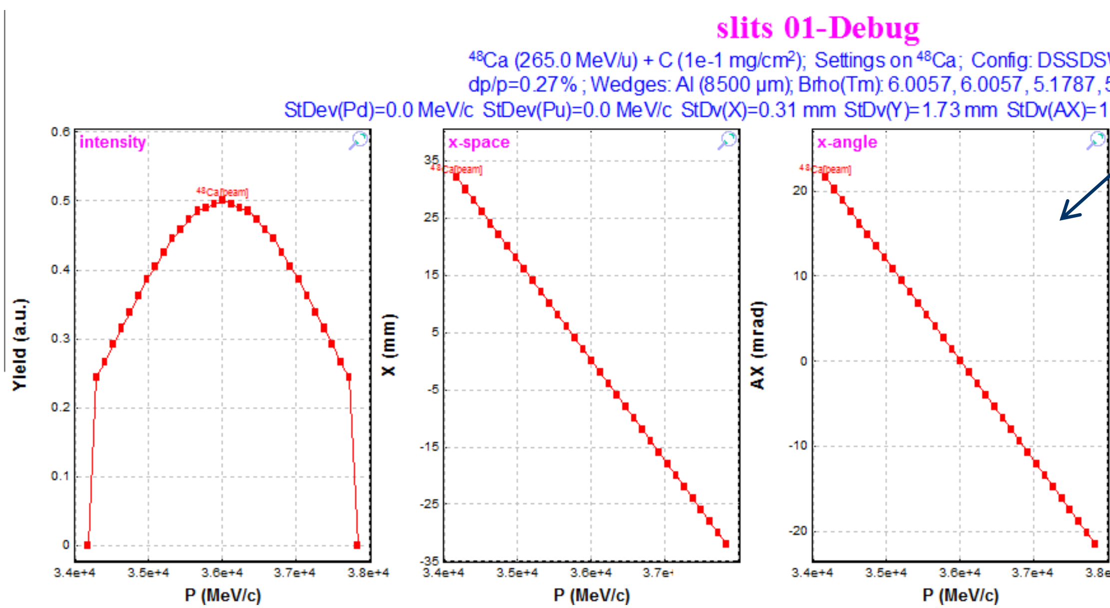




Cut of distributions after passing spatial slits







Cut of Initial distributions after passing slits

Benchmarks have been done with the new version with a beam inclination using matrices with non-zero x/d, t/x, and t/d elements.

Projectile fragmentation case: http://groups.nscl.msu.edu/lise/9_2/9_2_2/9_2_2.pdf

Abrasion-Fission case: http://groups.nscl.msu.edu/lise/9_2/9_2_5/9_2_5.pdf

Comparison from Mauricio : http://groups.nscl.msu.edu/lise/9_2/9_2_7/20100819_FRIB_discussion1.pdf

The new version should be checked more for some specific cases as

- *huge values of some optical matrix elements*
- *with all LISE++ blocks*
- *with all reactions in LISE++*

etc