

NOTE: Userlib_hys tools functions on the control computer in the PyDas shell. However please note that the PyDas shell version is behind in capabilities to the Analysis PC version. It is recommended that the Analysis computer be used with these planning tools.

Tutorial for current userlib_hys tools:

1. Setting up on Analysis PC from a terminal window

```
cd /SNS/users/inelastic/HYSPEC
ipython
run -i rotations.py
run -i ubmatrix_out.py
run -i userlib_hys_out.py
```

2. Useful on all DGS instruments

- a. Converters

- i. E2V(E_meV): output is velocity in m/s
- ii. V2E(V): output is energy in meV
- iii. Lambda2E(Lambda)
- iv. E2Lambda(E)
- v. E2K(E)
- vi. K2E(K)

- b. Q_lab: inputs are Ei (meV) and 2theta (deg), outputs are components of Q with z along ki

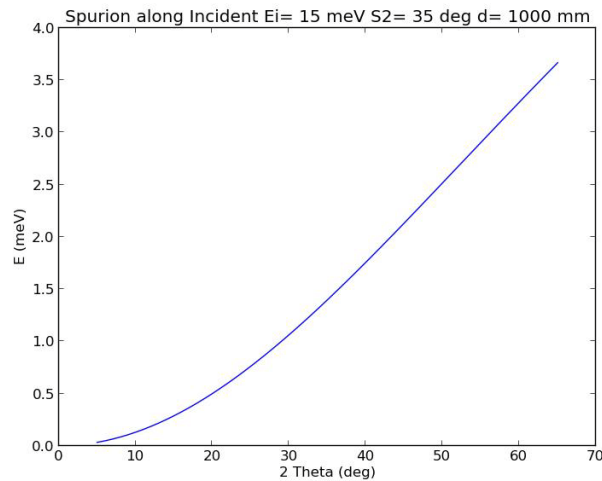
```
Q_lab(25, -35)
Q_lab(25, 35, E_meV=5)

output: [-3.1, 0, 0.93]
```

c. Spurion Calculators

- i. A parabolic energy spurion that may have some aluminum Powder points, arising from scatter upstream or downstream of the sample, somewhere along the monochromatic incident beam. $D_{\text{along_mm}}$ can be positive or negative, causing an energy loss or energy gain feature, respectively

```
Plot_Spurion_along_incident(Ei,S2,d_along_mm)
```



```
Plot_Spurion_along_incident(15,35,1000)
```

- ii. Spurion from isotropic scatter from sample position, scattering off of something cylindrically symmetric like our OVC, then back to detectors. Output is flat energy feature in meV.

```
Spurion_fromSampleNBackscatter(Ei,r_mm)  
Spurion_fromSampleNBackscatter(15,500)  
output: flat energy spurion at 4.95 meV
```

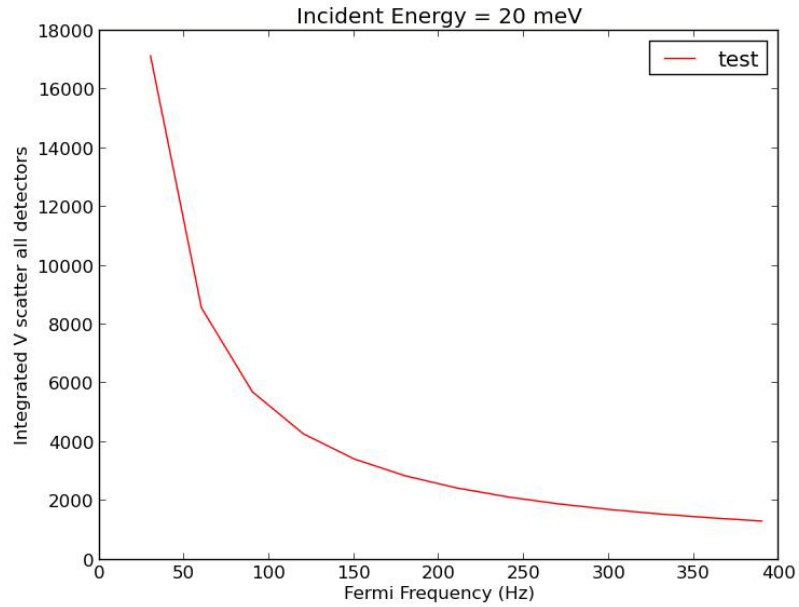
- iii. Spurion from isotropic scatter from sample position, scattering directly above or below, then back to detectors. Output is flat energy feature in meV.

```
Spurion_fromVerticalOffscatter(Ei,r_mm)  
Spurion_fromVerticalOffscatter(15,60)  
output: flat energy spurion at 0.39 meV
```

3. HYSPEC-specific, simple

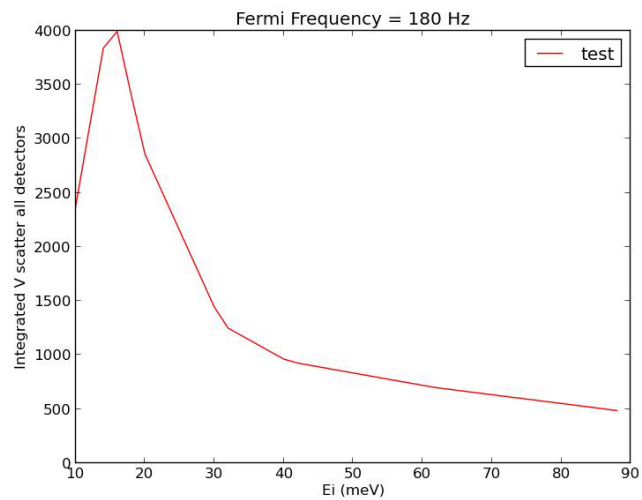
a. Flux

```
#flux_plot_Fermi(Ei_meV)
```



```
flux_plot_Fermi(20.0)
```

```
#flux_plot_Ei(Fermi_Hz)
```



```
flux_plot_Ei(180)
```

```
#relative_flux(Ei1_meV,Fermi1_Hz,Ei2_meV,Fermi2_Hz)
```

```
relative_flux(20,180,20,420)
```

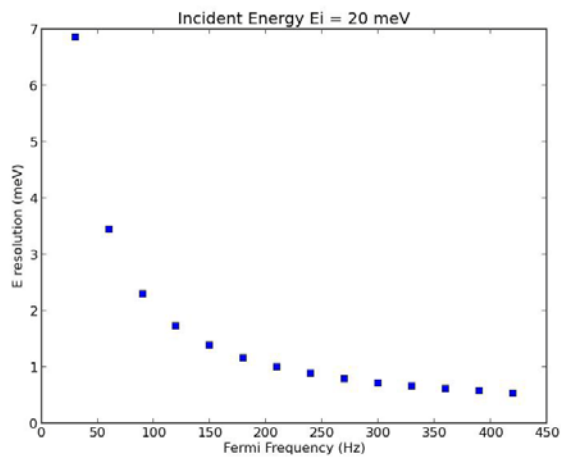
```
output: relative flux ratio is 2.3
```

```
#V_flux(Ei_meV,Fermi_Hz)
V_flux(20,180)
output: flux from scatter from vanadium rod in arbitrary
units is 2858
```

b. Energy Resolution

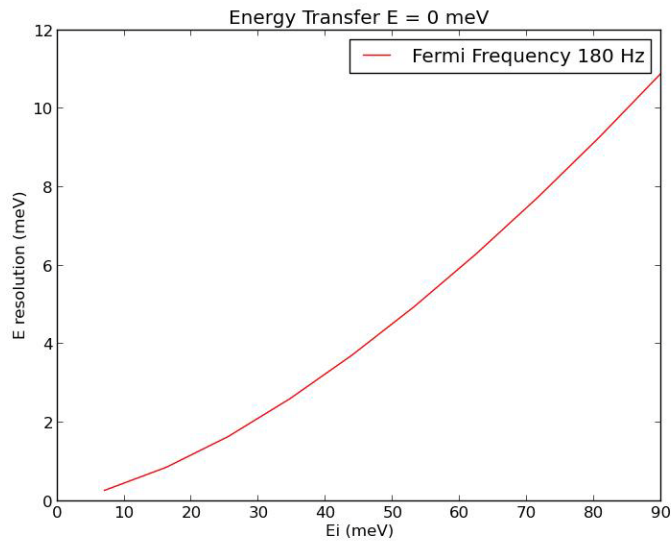
```
#Resolution_Calc_single(Ei,Ef,msd=1.8,FermiFreq=180.0)
Resolution_Calc_single(20,20)
output: energy resolution in FWHM is 1.16 meV
```

```
#Plot_Resolution_vs_Fermi(Ei,E=0,msd=1.8)
```



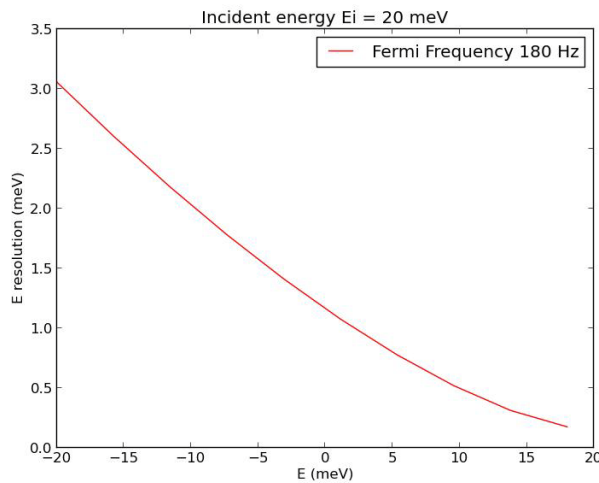
```
Plot_Resolution_vs_Fermi(20.0)
```

```
#Plot_Resolution_vs_Ei(EiMin,EiMax,nEi,E=0,msd=1.8,FermiFre
q=[180.0])
```



```
Plot_Resolution_vs_Ei(7,90,10)
```

```
#Plot_Resolution_vs_E(Ei,EMin,EMax,nE,msd=1.8,FermiFreq=[180.0])
```



```
Plot_Resolution_vs_E(20,-20,18,10)
```

```
#Plot_Resolution_vs_Ef(Ei,EfMin,EfMax,nEf,msd=1.8,FermiFreq=[180.0])
```

not working right now...

```
Plot_Resolution_vs_Ef(20,2,80,10)
```

c. Q resolution

- i. *Not done yet. Delta Kf comes from Ef, while Delta transverse comes from detector tube width in angle. Others prefer to do McStas models, but for a foot in the door...*

d. E range / Prompt Pulse impact

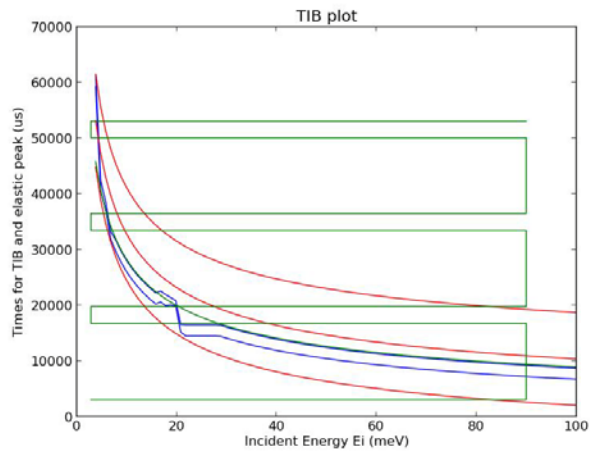
i. Simple:

```
#IsEiOverPromptPulse(Ei)
IsEiOverPromptPulse(20)
output: good Ei
```

ii. Fancier

```
SpurionPromptPulse(Ei_meV,msd=1800,tail_length_us=3000
,talk=True)
SpurionPromptPulse(20)
output: ????
```

iii. Plotting impact

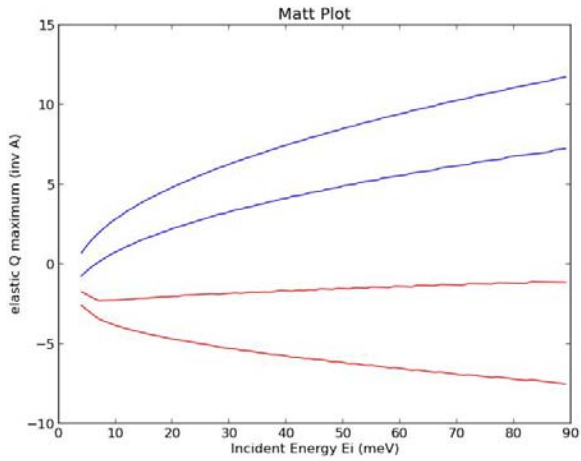


```
PlotTIB()
```

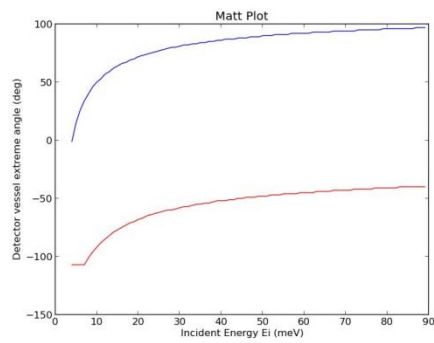
e. Q range

```
ExtremeElasticQ(Ei_meV,foc_option='PG',**kwargs)
ExtremeElasticQ(20)
output: ????
```

```
PlotExtremeQ(S2instead=False,M2Xaxis=False)
```



`PlotExtremeQ()`



`PlotExtremeQ(S2instead=True)`

`PlotExtremeQ(M2Xaxis=True)` *not working yet*

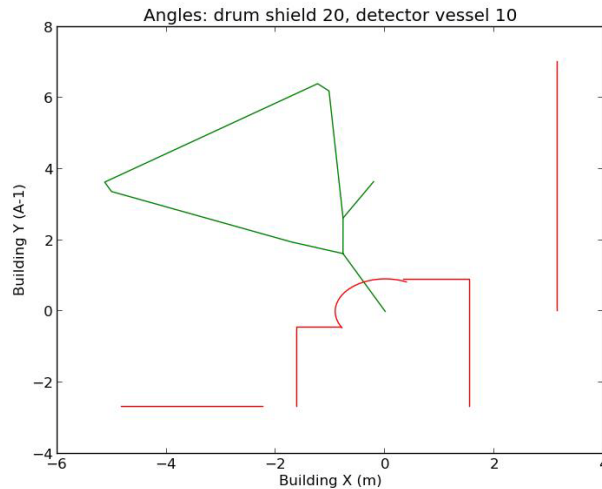
- f. Calculators for Fermi phase, M2 angle, etc.

`Fermi_Phase(Ei_meV, Fermi_Hz)`

- g. Collision avoidance tools

- i. Similar tools wrapped into instrument class below
- ii. Inputs: M2 (drum shield angle in degrees), S2 (detector vessel angle in degrees)

```
#Real_space_plot(M2_deg_try, S2_deg_try,
MSD_m_try=1.8, SafetyRadius_m=0.25, FillStation3He=0 )
```



```
Real_space_plot(20,10)
```

```
Is_Position_Safe(15,15)
#Is_Position_Safe(M2_deg_try, S2_deg_try,
MSD_m_try=1.8, SafetyRadius_m=0.25, FillStation3He=0 )
```

4. Single Crystal class

- a. *There's more...*
- b. May be useful for other DGS instruments
- c. Bundles OrientedLattice, Peak (ISAW format) data, and other crystal specific info.
Functions have already translated angles to 2Theta with a sign and sample rotation about a vertical axis normal to scattering plane.
- d. Initialization

NOTE the example below follows the respective format:

```
CubicCrystal=Crystal((a,b,c),(h1,k1,l1),(h2,k2,l2))
```

```
hexCrystal=Crystal((a,b,c),(h1,k1,l1),(h2,k2,l2),alphanbetagamma(alpha,
beta,gamma))
```

- i. inputs are (a,b,c) in Angstroms, and two hkl (rlu) vectors that define a nominal plane, which for HYSPEC is the horizontal, scattering plane

```
CubicCrystal=Crystal((4,4,4),(1,0,0),(0,1,0))
```

```
hexCrystal =
Crystal((4,4,4),(1,0,0),(0,1,0),alphanbetagamma=(90.0,9
0.0,120.0))
```


- e. Angle between two hkl vectors

```
CubicCrystal.Angle((h1,k1,l1),(h2,k2,l2))
CubicCrystal.Angle((1,0,0),(0,1,0))
```

- f. Finding 2-theta and theta for an hkl, E combination

- i. inputs are hkl (rlu) and Ei (meV); outputs are 2Theta and sample angle *and real sample angle (not yet)*

```
CubicCrystal.WhereIsBragg((1,0,0),25)
CubicCrystal.WhereIsBragg((1,0,0),25,left=True)
```

```
Crystal.WhereIsBragg(hkl,Ei_meV,E_meV=0.0,left=False,
S1offset=0.0)
```

5. Powder class

- a. Doesn't exist yet...

6. HYSPEC Instrument class CheckOutMotors

- a. *There's more...*

- b. Allows one to pass as a parameter the entire instrument configuration, instead of specific Ei's, angles, etc.

- c. Initialization

- i. In the Analysis cluster this provides some default values for instrument motors, etc.

```
trialconfig=CheckOutMotors()
s1 (sample vertical axis rotation) (deg): 5
s2 (detector vessel mid-angle) (deg): 30
Ei (meV): 20
```

- ii. On the control computer, initialization takes in actual values from process variables.

- d. Q / 2Theta / d spacing calculator

- i. Assumes elastic (E=0)
- ii. Outputs:

```
(Q (inv A), d (Angstrom), 2Theta (deg), visible in detectors?)
```

- iii. Starts with any one of these inputs and generates the rest

```
trialconfig.fr2Theta(3)
```

```
trialconfig.frQ(3)
trialconfig.frd(3)
```

- e. Determining the angle a particular detector tube is at, given S2, OR determining S2 from a detector index and angle

```
trialconfig.theta_for_S2_det()
trialconfig.S2_for_theta_det(25)
trialconfig.det_tube = 3
trialconfig.theta_for_S2_det()
trialconfig.S2_for_theta_det(25)
trialconfig.det_tube = 80
```

- f. Functions similar to real space calculators above

```
trialconfig.PlotRealSpace()
trialconfig.DetectorVesselSafe()
```

- g. Determining M2 for a desired Ei, knowing whether one is using PG or Heusler focusing element.

```
trialconfig.M2calc(33)
```

- h. Note: trialconfig.UseCurrent isn't available on ANALYSIS, but with PYDAS it sets to the current actual settings for those components listed, using a format similar to CheckParmCheck

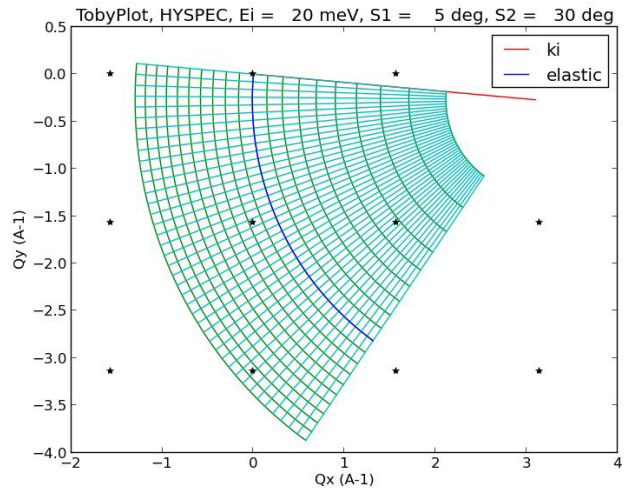
```
trialconfig.CheckParmCheck(['s1', 's2', 'sgu', 'sgl', 'm2', 'vMe1', 'Ei', 'msd', 'FermiFreq', 'det_tube'])
```

7. Tools that use both Crystal Class and CheckOutMotors class

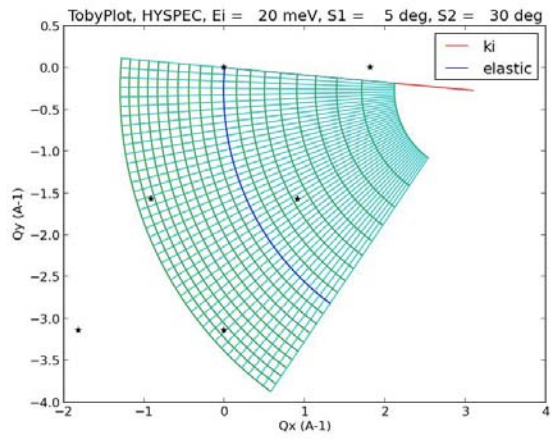
- a. Alignment Plot, monochromatic, multi rotation angle (Q-ring user friendly)
 - i. *Not complete...*
- b. TobyPlot
 - i. Parameters:

```
Crystal class, CheckOutMotors class, hkl=(0,0,0),
Echoice=0.0, Emin_meV=0.0, Emax_meV=0.0, Edelta=2.0,
DetStep=4
```

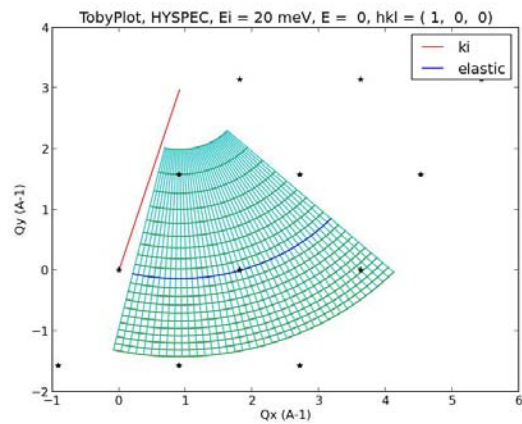
ii. Examples



`TobyPlot(CubicCrystal,trialconfig)`

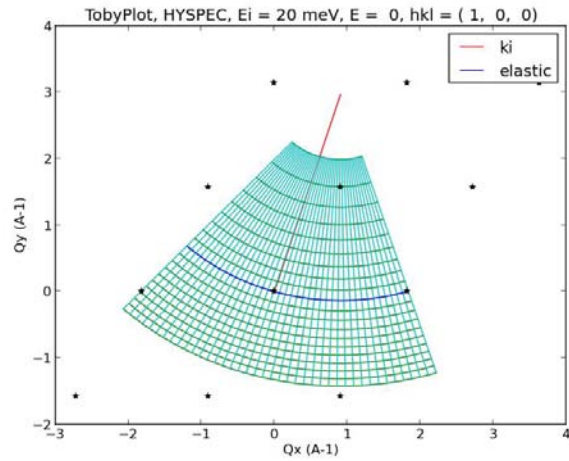


`TobyPlot(hexCrystal,trialconfig)`



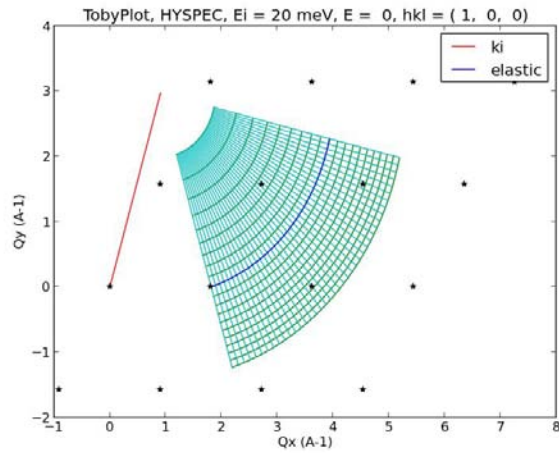
`TobyPlot(hexCrystal,trialconfig,hkl=(1,0,0))`

`trialconfig.det_tube=1`

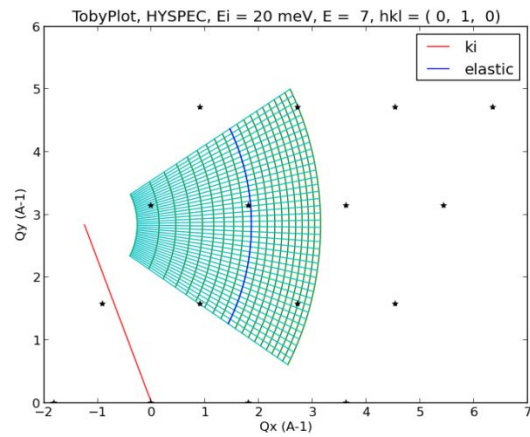


`TobyPlot(hexCrystal,trialconfig,hkl=(1,0,0))`

`trialconfig.det_tube=160`



`TobyPlot(hexCrystal,trialconfig,hkl=(1,0,0))`



`TobyPlot(hexCrystal,trialconfig,hkl=(0,1,0),Echoice=7.`

0)

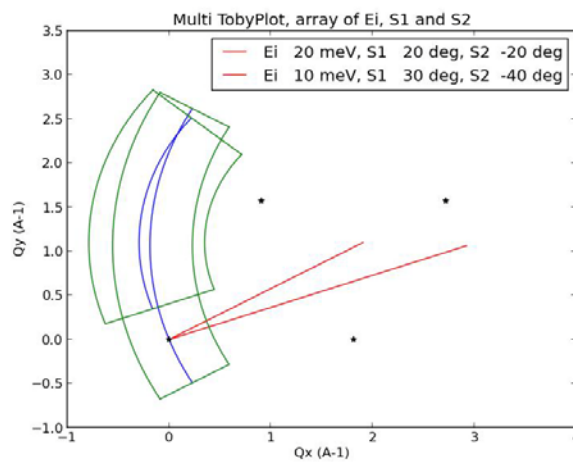
iii. Need an option where hkl doesn't change $S2$ but does confirm that it's in range...

c. MultiTobyPlot:

- i. for scan planning, it helps to have a skeletal version of TobyPlot replicated throughout the screen
- ii. Parameters:

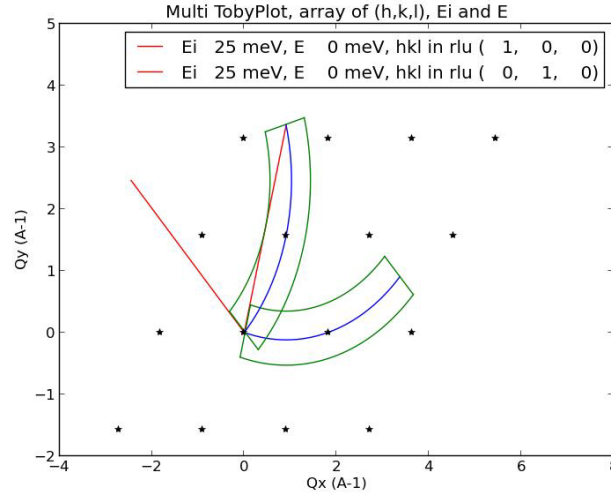
```
Crystal class, CheckOutMotors class, EiS1S2_array=[],  
hklEiE_array=[], Emin=0.0, Emax=0.0
```

- iii. Input option 1: but E_i (meV), $S1$ sample v -axis rotation angle (deg), and $S2$ detector vessel angle (deg)



```
ManyTobyPlots(hexCrystal, trialconfig, EiS1S2_array =  
[(20, 20, -20), (10, 30, -40)])
```

iv. Input option 2: h, k, l (all rlu), Ei (meV), E (meV)



```
ManyTobyPlots(hexCrystal,trialconfig,hklEiE_array =
  [(1,0,0,25,0),(0,1,0,25,0)])
```

v. Input option 3: like Input option 2, but fixing S2 instead of driving S2 to the desired detector / 2Theta. Does confirm that 2Theta is in range for different configurations.

1. *Not started...*

d. Plotting E as a function of Qx for a given Qy, or vica verca.

i. *Not complete...*

8. Usable only on PYDAS on control computer for HYSPEC

a. Parameter check

```
parm_check(['s1','mlpg'])
#parm_check(parmlist) -- real value check
```

b. set_Ei

```
set_Ei(25)
set_Ei(25,Fermi_Hz=180) #optional change of frequency
set_Ei(25,test=1) #crunches numbers but doesn't change any
process variables
set_Ei(25,flat=1) #flattens the focusing element
#set_Ei(Ei_meV,Fermi_Hz=0.0,etol=0.01,ftol=3.0,flat=0,test=
0)
```

c. wiggleS1_run

i. *Not done yet*

d. Buffer command system used at Sequoia

i. *Not done yet.*

9. File management and recording

a. Generating a list of files from the analysis cluster from a terminal command line

```
#nxsummarytable -i HYS run-number >filename.txt  
nxsummarytable -i HYS 100-200,300-400
```

b. Mounting your analysis user disk area from the control PC:

- i. Open Computer from Start Icon on lower left
- ii. Choose 'map network drive'
- iii. Type in '123.45.67.89\abc' where 'abc' should be changed to your ucams or xcams ID
- iv. Click on 'finish' and a window will pop up for you to log in
- v. Type in 'ornl\abc' where 'abc' should be changed to your ucams or xcams ID
- vi. Type in your ucams or xcams password
- vii. Click OK and a new directory window will pop up.
- viii. Move files to your account directory as needed.

c. Dismounting (please do this as soon as possible for your own account security, since all have access to the control PC)

- i. Close all relevant account windows.
- ii. Open Computer from Start Icon on lower left
- iii. Highlight your mounted network disk, that shows your ucams or xcams ID
- iv. Right-click and choose 'Disconnect'. Your mounted disk ought to disappear. If not, a window will pop up with useful guidance related to not doing step i correctly.

d. Copying home-made logs from PYDAS shell

- i. Highlight relevant text under shell tab of PYDAS
- ii. NEVER use control-C. That stops runs as well as copies.
- iii. Instead, right-click on the highlighted text and choose 'Copy'
- iv. Paste into something like a 'notepad' window
- v. Print, save, or whatever...

e. Good practice from PYDAS shell is to put your scripts and scan csv and files into a subdirectory of PydasLogs

- i. Choose shell tab
- ii. Type 'pwd' to confirm that you aren't in a subdirectory already. If you are, type 'cd ..'
- iii. Type 'mkdir IPTS-<number>'
- iv. Type 'cd IPTS-<number>'

10. M-SLICE

- a. On analysis cluster, create a terminal window and enter

dave-dev

- b. On main screen, from the tab row at top, click on:

- i. Data Reduction

1. NCNR

- a. DGS Reduction (Mslice)
- b. Choose *.nxspe format

- c. On Mslice screen

- i. Sample type:

1. Single Crystal

- a. Diffuse Scattering Multiple E

- ii. Energy Range

1. All

- iii. Option

1. View Axis

- a. Allow Extra Viewing Axis

- iv. Option

1. Binning Method

- a. Constant Intervals

- i. Yes