Updated on Nov. 20, 2020

Installation

- Get the folder “sixcircle_ver1.3alpha_xxxxxxxx”.
- It includes:

sixcircle.py The main program.

scbasic.py A required module of six-circle calculation. If necessary, users can develop their own six-circle program based on scbasic.py.

ini.conf A configuration file essential for initialization

sixcircle_rqd.py An optional module specially for BL43LXU and BL35XU at SPring-8. It is also an example to show how users can develop their own toolkits based on sixcircle.py

BL43XU_CONST.mac A configuration file including constants in BL43LXU, SPring8, required by sixcircle_rqd.py

BL35XU_CONST.mac A configuration file including constants in BL35XU, SPring8, required by sixcircle_rqd.py

- The program relies on python3, numpy, and scipy.

```
pip install numpy
pip install scipy
```

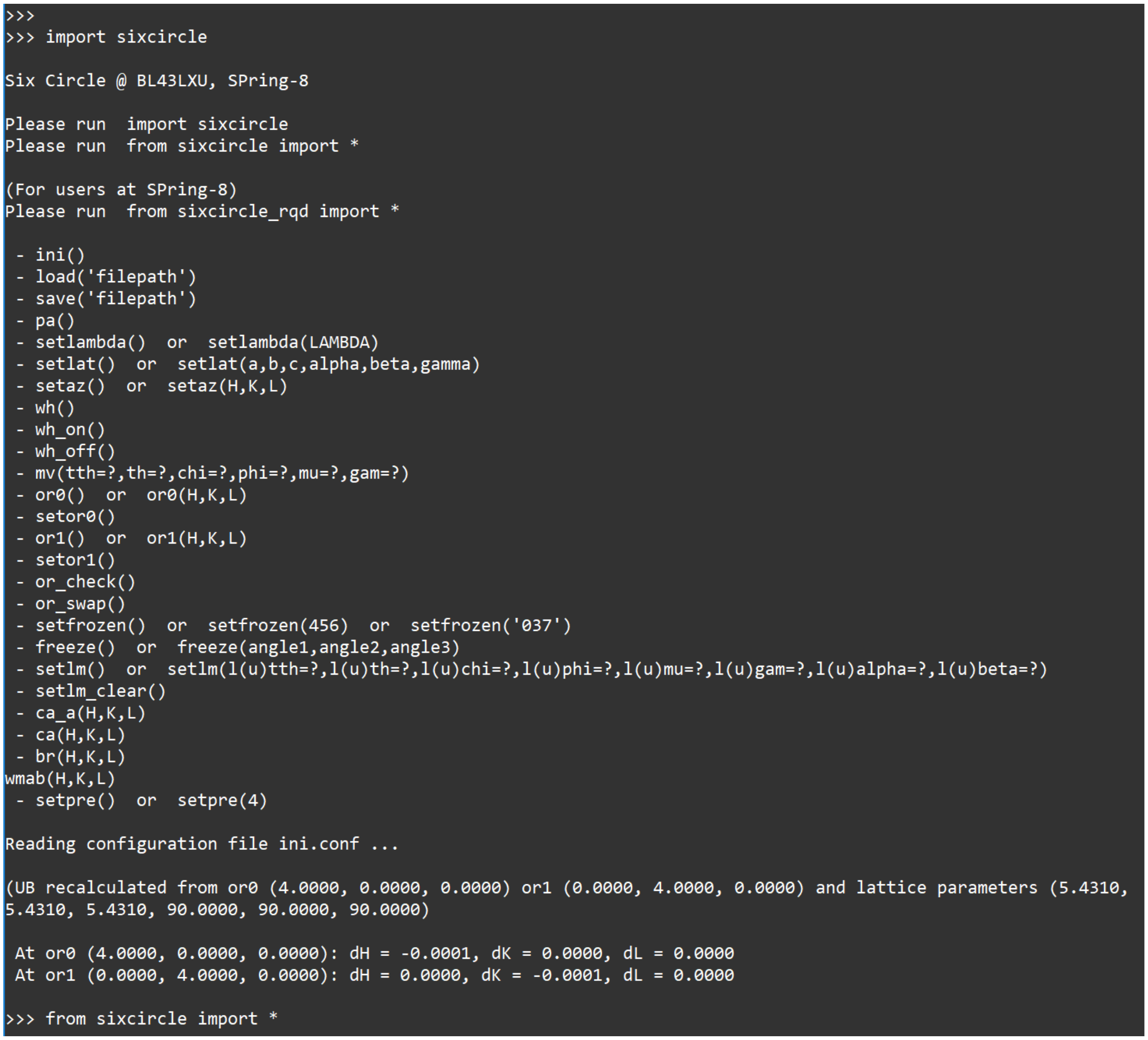
Start up

Get into the folder “sixcircle_ver1.4alpha_xxxxxxxx” and enter python3 shell.

```
import sixcircle
from sixcircle import *
```

For users at BL43LXU, add:

```
from sixcircle_rqd import *
```



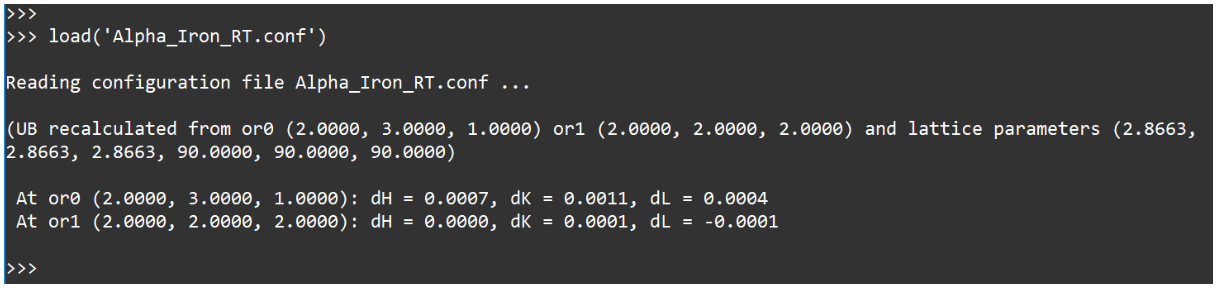
```
>>>
>>> import sixcircle

Six Circle @ BL43LXU, SPring-8

Please run  import sixcircle
Please run  from sixcircle import *

(For users at SPring-8)
Please run  from sixcircle_rqd import *

 - ini()
 - load('filepath')
 - save('filepath')
 - pa()
 - setlambda()  or  setlambda(LAMBDA)
 - setlat()  or  setlat(a,b,c,alpha,beta,gamma)
 - setaz()  or  setaz(H,K,L)
 - wh()
 - wh_on()
 - wh_off()
 - mv(tth=?,th=?,chi=?,phi=?,mu=?,gam=?)
 - or0()  or  or0(H,K,L)
 - setor0()
 - or1()  or  or1(H,K,L)
 - setor1()
 - or_check()
 - or_swap()
 - setfrozen()  or  setfrozen(456)  or  setfrozen('037')
 - freeze()  or  freeze(angle1,angle2,angle3)
 - setlm()  or  setlm(l(u)tth=?,l(u)th=?,l(u)chi=?,l(u)phi=?,l(u)mu=?,l(u)gam=?,l(u)alpha=?,l(u)beta=?)
 - setlm_clear()
 - ca_a(H,K,L)
 - ca(H,K,L)
 - br(H,K,L)
wmab(H,K,L)
 - setpre()  or  setpre(4)

Reading configuration file ini.conf ...

(UB recalculated from or0 (4.0000, 0.0000, 0.0000) or1 (0.0000, 4.0000, 0.0000) and lattice parameters (5.4310,
5.4310, 5.4310, 90.0000, 90.0000, 90.0000)

 At or0 (4.0000, 0.0000, 0.0000): dH = -0.0001, dK = 0.0000, dL = 0.0000
 At or1 (0.0000, 4.0000, 0.0000): dH = 0.0000, dK = -0.0001, dL = 0.0000

>>> from sixcircle import *
```

Definition of motors

mu:	right-hand rotation angle of incident beam with respect to x-axis.
gam:	right-hand rotation angle of deflected beam with respect to x-axis.
sa:	scattering angle
omega:	difference from tth/2 to th

load('filepath')

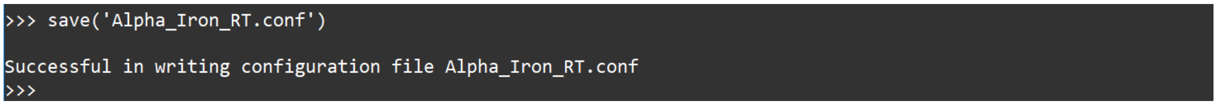
- Load a configuration file



```
>>>
>>> load('Alpha_Iron_RT.conf')

Reading configuration file Alpha_Iron_RT.conf ...

(UB recalculated from or0 (2.0000, 3.0000, 1.0000) or1 (2.0000, 2.0000, 2.0000) and lattice parameters (2.8663,
2.8663, 2.8663, 90.0000, 90.0000, 90.0000)

 At or0 (2.0000, 3.0000, 1.0000): dH = 0.0007, dK = 0.0011, dL = 0.0004
 At or1 (2.0000, 2.0000, 2.0000): dH = 0.0000, dK = 0.0001, dL = -0.0001

>>>
```

save('filepath')

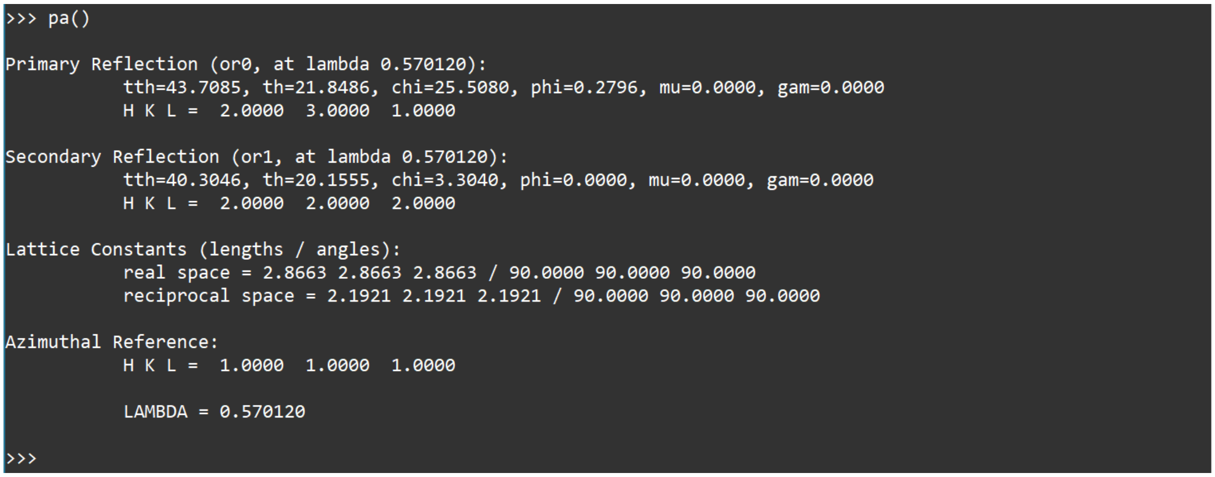
- Save to a configuration file



```
>>> save('Alpha_Iron_RT.conf')

Successful in writing configuration file Alpha_Iron_RT.conf
>>>
```

pa()

- Check current parameters

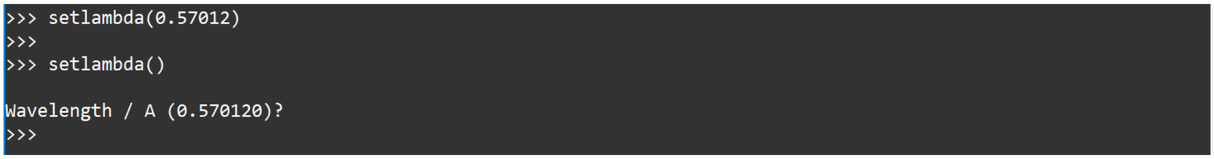


```
>>> pa()

Primary Reflection (or0, at lambda 0.570120):
           tth=43.7085, th=21.8486, chi=25.5080, phi=0.2796, mu=0.0000, gam=0.0000
           H K L =  2.0000  3.0000  1.0000

Secondary Reflection (or1, at lambda 0.570120):
           tth=40.3046, th=20.1555, chi=3.3040, phi=0.0000, mu=0.0000, gam=0.0000
           H K L =  2.0000  2.0000  2.0000

Lattice Constants (lengths / angles):
           real space = 2.8663 2.8663 2.8663 / 90.0000 90.0000 90.0000
           reciprocal space = 2.1921 2.1921 2.1921 / 90.0000 90.0000 90.0000

Azimuthal Reference:
           H K L =  1.0000  1.0000  1.0000

           LAMBDA = 0.570120

>>>
```

setlambda()

setlambda(LAMBDA)

- Set wavelength

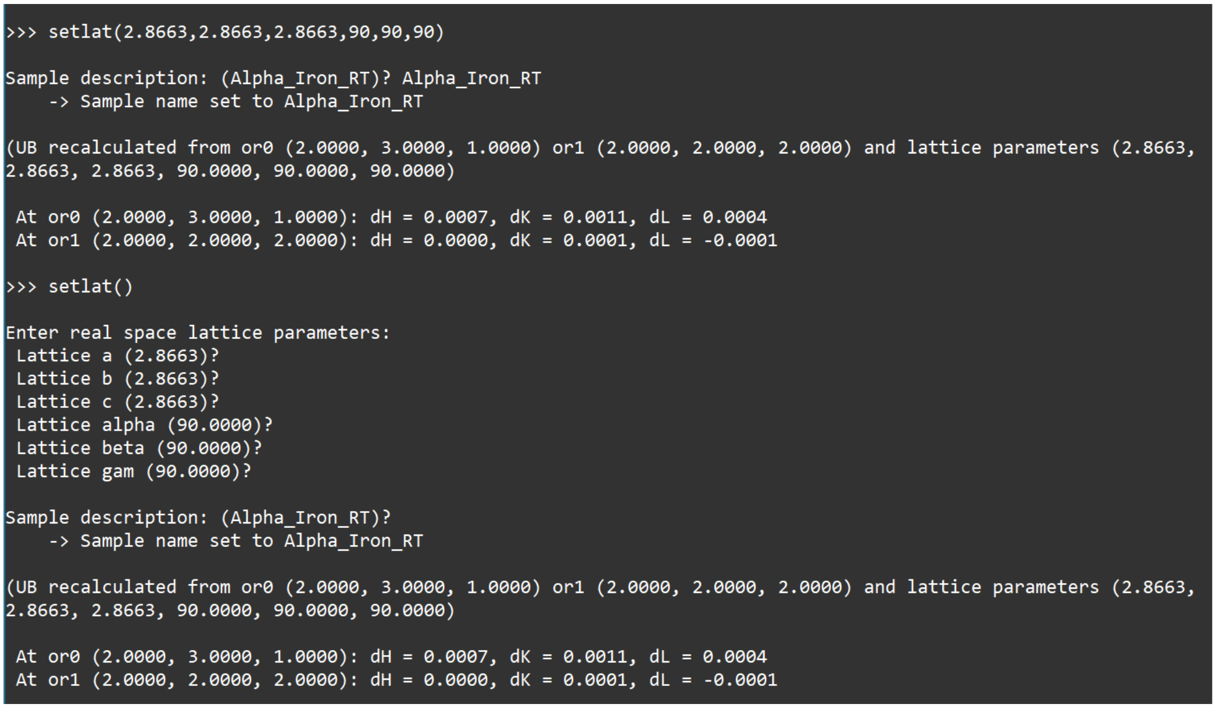


```
>>> setlambda(0.57012)
>>>
>>> setlambda()

Wavelength / A (0.570120)?
>>>
```

setlat()

setlat(a,b,c,alpha,beta,gamma)

- Set lattice parameters

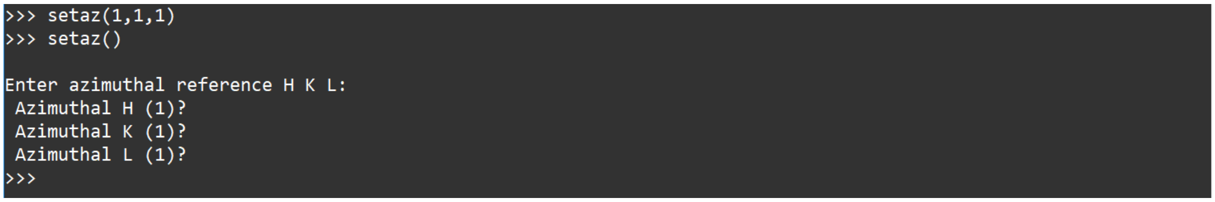


```
>>> setlat(2.8663,2.8663,2.8663,90,90,90)

Sample description: (Alpha_Iron_RT)? Alpha_Iron_RT
    -> Sample name set to Alpha_Iron_RT

(UB recalculated from or0 (2.0000, 3.0000, 1.0000) or1 (2.0000, 2.0000, 2.0000) and lattice parameters (2.8663,
2.8663, 2.8663, 90.0000, 90.0000, 90.0000)

 At or0 (2.0000, 3.0000, 1.0000): dH = 0.0007, dK = 0.0011, dL = 0.0004
 At or1 (2.0000, 2.0000, 2.0000): dH = 0.0000, dK = 0.0001, dL = -0.0001

>>> setlat()

Enter real space lattice parameters:
 Lattice a (2.8663)?
 Lattice b (2.8663)?
 Lattice c (2.8663)?
 Lattice alpha (90.0000)?
 Lattice beta (90.0000)?
 Lattice gam (90.0000)?

Sample description: (Alpha_Iron_RT)?
    -> Sample name set to Alpha_Iron_RT

(UB recalculated from or0 (2.0000, 3.0000, 1.0000) or1 (2.0000, 2.0000, 2.0000) and lattice parameters (2.8663,
2.8663, 2.8663, 90.0000, 90.0000, 90.0000)

 At or0 (2.0000, 3.0000, 1.0000): dH = 0.0007, dK = 0.0011, dL = 0.0004
 At or1 (2.0000, 2.0000, 2.0000): dH = 0.0000, dK = 0.0001, dL = -0.0001
```

setaz()

setaz(H, K, L)

- Set the reference vector HKL

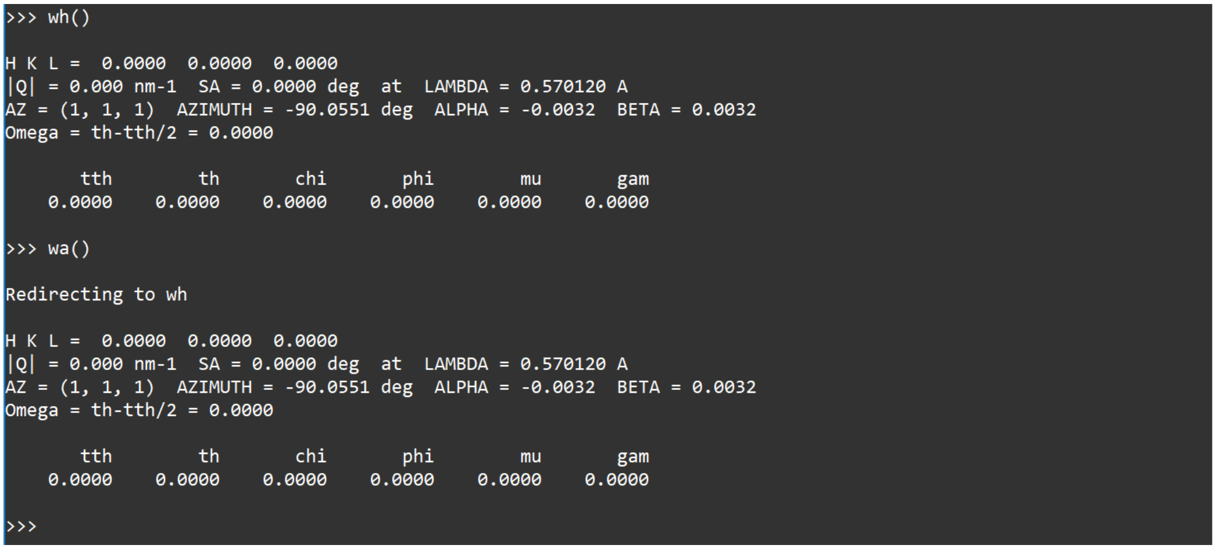


```
>>> setaz(1,1,1)
>>> setaz()

Enter azimuthal reference H K L:
 Azimuthal H (1)?
 Azimuthal K (1)?
 Azimuthal L (1)?
>>>
```

wh()

wa()

- Check all current positions

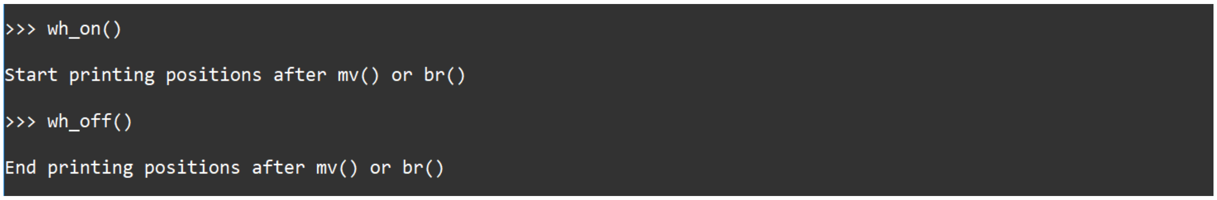


```
>>> wh()

H K L =  0.0000  0.0000  0.0000
|Q| = 0.000 nm-1  SA = 0.0000 deg  at  LAMBDA = 0.570120 A
AZ = (1, 1, 1)  AZIMUTH = -90.0551 deg  ALPHA = -0.0032  BETA = 0.0032
Omega = th-tth/2 = 0.0000

       tth        th       chi       phi        mu       gam
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

>>> wa()

Redirecting to wh

H K L =  0.0000  0.0000  0.0000
|Q| = 0.000 nm-1  SA = 0.0000 deg  at  LAMBDA = 0.570120 A
AZ = (1, 1, 1)  AZIMUTH = -90.0551 deg  ALPHA = -0.0032  BETA = 0.0032
Omega = th-tth/2 = 0.0000

       tth        th       chi       phi        mu       gam
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

>>>
```

wh_on()

wh_off()

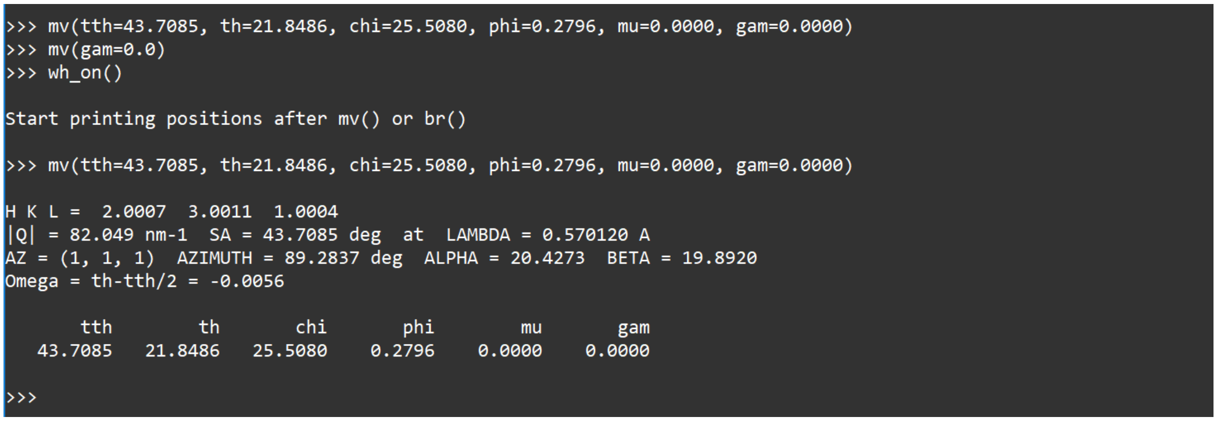
- Start (end) printing positions after mv() or br()



```
>>> wh_on()

Start printing positions after mv() or br()

>>> wh_off()

End printing positions after mv() or br()
```

mv(tth=?, th=?, chi=?, phi=?, mu=?, gam=?)

- Move motors positions.



```
>>> mv(tth=43.7085, th=21.8486, chi=25.5080, phi=0.2796, mu=0.0000, gam=0.0000)
>>> mv(gam=0.0)
>>> wh_on()

Start printing positions after mv() or br()

>>> mv(tth=43.7085, th=21.8486, chi=25.5080, phi=0.2796, mu=0.0000, gam=0.0000)

H K L =  2.0007  3.0011  1.0004
|Q| = 82.049 nm-1  SA = 43.7085 deg  at  LAMBDA = 0.570120 A
AZ = (1, 1, 1)  AZIMUTH = 89.2837 deg  ALPHA = 20.4273  BETA = 19.8920
Omega = th-tth/2 = -0.0056

       tth        th       chi       phi        mu       gam
   43.7085   21.8486   25.5080    0.2796    0.0000    0.0000

>>>
```

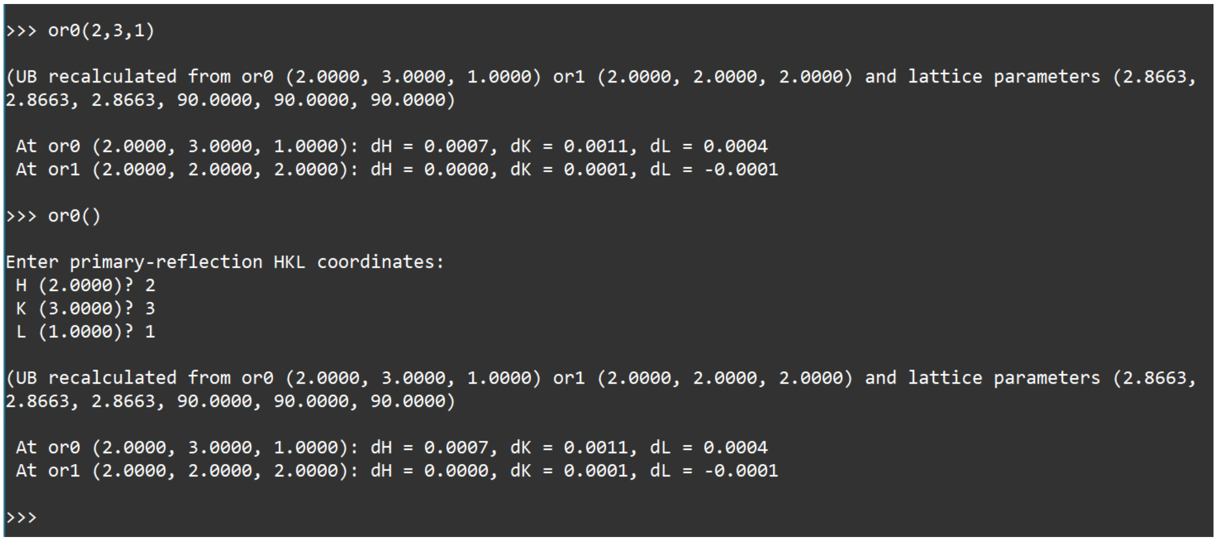
or0()

or0(H,K,L)

or1()

or1(H,K,L)

- Set the HKL of the primary (secondary) reflection based on current motors positions.

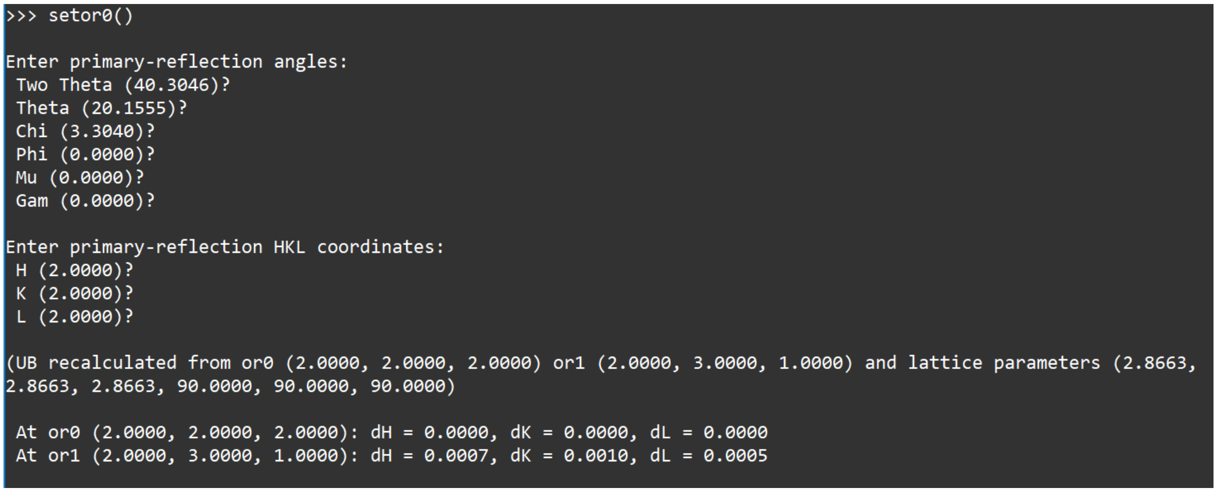


```
>>> or0(2,3,1)

(UB recalculated from or0 (2.0000, 3.0000, 1.0000) or1 (2.0000, 2.0000, 2.0000) and lattice parameters (2.8663,
2.8663, 2.8663, 90.0000, 90.0000, 90.0000)

 At or0 (2.0000, 3.0000, 1.0000): dH = 0.0007, dK = 0.0011, dL = 0.0004
 At or1 (2.0000, 2.0000, 2.0000): dH = 0.0000, dK = 0.0001, dL = -0.0001

>>> or0()

Enter primary-reflection HKL coordinates:
 H (2.0000)? 2
 K (3.0000)? 3
 L (1.0000)? 1

(UB recalculated from or0 (2.0000, 3.0000, 1.0000) or1 (2.0000, 2.0000, 2.0000) and lattice parameters (2.8663,
2.8663, 2.8663, 90.0000, 90.0000, 90.0000)

 At or0 (2.0000, 3.0000, 1.0000): dH = 0.0007, dK = 0.0011, dL = 0.0004
 At or1 (2.0000, 2.0000, 2.0000): dH = 0.0000, dK = 0.0001, dL = -0.0001

>>>
```

setor0()

setor1()

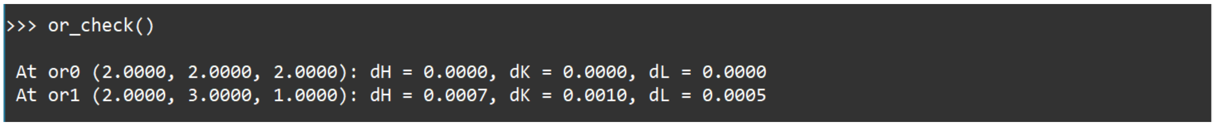
- Set the HKL of the primary (secondary) reflection and corresponding motors positions.



```
>>> setor0()

Enter primary-reflection angles:
 Two Theta (40.3046)?
 Theta (20.1555)?
 Chi (3.3040)?
 Phi (0.0000)?
 Mu (0.0000)?
 Gam (0.0000)?

Enter primary-reflection HKL coordinates:
 H (2.0000)?
 K (2.0000)?
 L (2.0000)?

(UB recalculated from or0 (2.0000, 2.0000, 2.0000) or1 (2.0000, 3.0000, 1.0000) and lattice parameters (2.8663,
2.8663, 2.8663, 90.0000, 90.0000, 90.0000)

 At or0 (2.0000, 2.0000, 2.0000): dH = 0.0000, dK = 0.0000, dL = 0.0000
 At or1 (2.0000, 3.0000, 1.0000): dH = 0.0007, dK = 0.0010, dL = 0.0005
```

or_check()

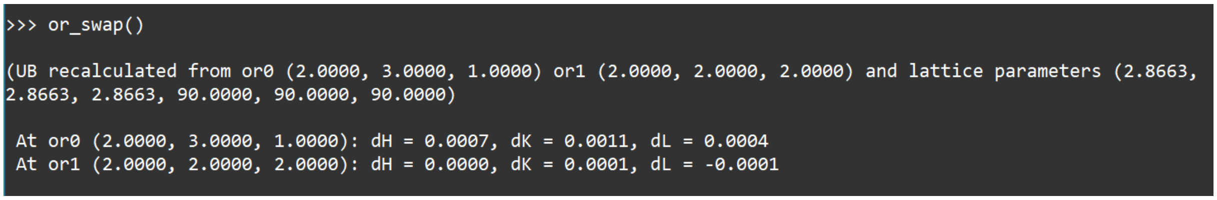
- Check consistency at the primary and the secondary reflections.



```
>>> or_check()

 At or0 (2.0000, 2.0000, 2.0000): dH = 0.0000, dK = 0.0000, dL = 0.0000
 At or1 (2.0000, 3.0000, 1.0000): dH = 0.0007, dK = 0.0010, dL = 0.0005
```

or_swap()

- Swap the primary reflection and the secondary reflection.



```
>>> or_swap()

(UB recalculated from or0 (2.0000, 3.0000, 1.0000) or1 (2.0000, 2.0000, 2.0000) and lattice parameters (2.8663,
2.8663, 2.8663, 90.0000, 90.0000, 90.0000)

 At or0 (2.0000, 3.0000, 1.0000): dH = 0.0007, dK = 0.0011, dL = 0.0004
 At or1 (2.0000, 2.0000, 2.0000): dH = 0.0000, dK = 0.0001, dL = -0.0001
```

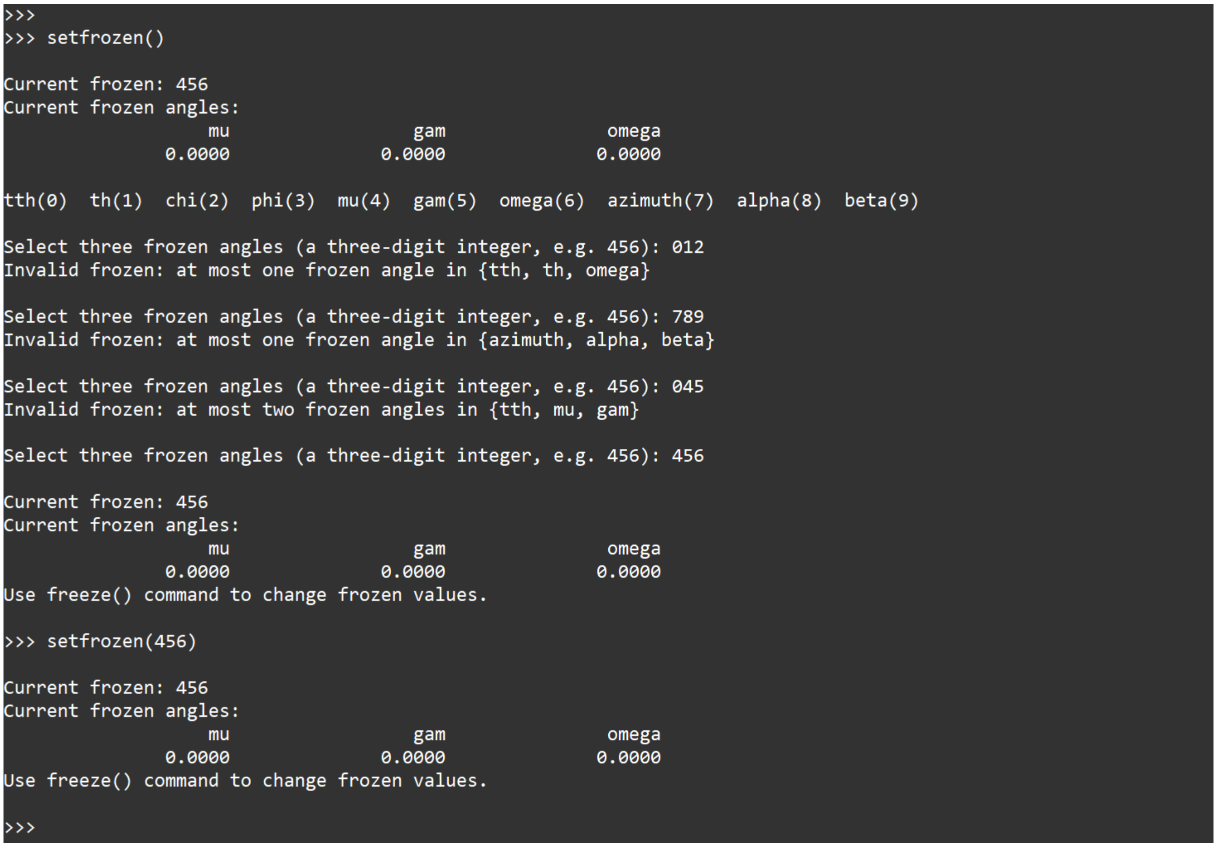
setfrozen()

setfrozen(456)

setfrozen(‘037’)

setmode()

- ‘Frozen’ (abc) is a string of a three-digit number.
- Choose three angles to freeze: tth (0), th (1), chi (2), phi (3), mu(4), gam (5), omega (6), azimuth (7), alpha (8), beta (9)
- Default mode is 456: freeze mu, gam, omega
- It is possible to choose one or two angles in {tth, mu, gam}
- It is possible to choose zero or one angle in {azimuth, alpha, beta}
- It is possible to choose zero or one angle in {tth, th, omega}

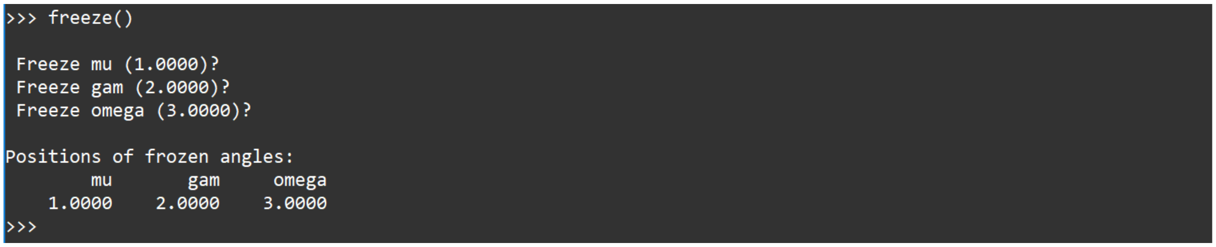


```
>>>
>>> setfrozen()

Current frozen: 456
Current frozen angles:
                   mu                 gam                omega
               0.0000              0.0000               0.0000

tth(0)  th(1)  chi(2)  phi(3)  mu(4)  gam(5)  omega(6)  azimuth(7)  alpha(8)  beta(9)

Select three frozen angles (a three-digit integer, e.g. 456): 012
Invalid frozen: at most one frozen angle in {tth, th, omega}

Select three frozen angles (a three-digit integer, e.g. 456): 789
Invalid frozen: at most one frozen angle in {azimuth, alpha, beta}

Select three frozen angles (a three-digit integer, e.g. 456): 045
Invalid frozen: at most two frozen angles in {tth, mu, gam}

Select three frozen angles (a three-digit integer, e.g. 456): 456

Current frozen: 456
Current frozen angles:
                   mu                 gam                omega
               0.0000              0.0000               0.0000
Use freeze() command to change frozen values.

>>> setfrozen(456)

Current frozen: 456
Current frozen angles:
                   mu                 gam                omega
               0.0000              0.0000               0.0000
Use freeze() command to change frozen values.

>>>
```

freeze()

freeze(position1,position2,position3)

- Set positions of three frozen angles.

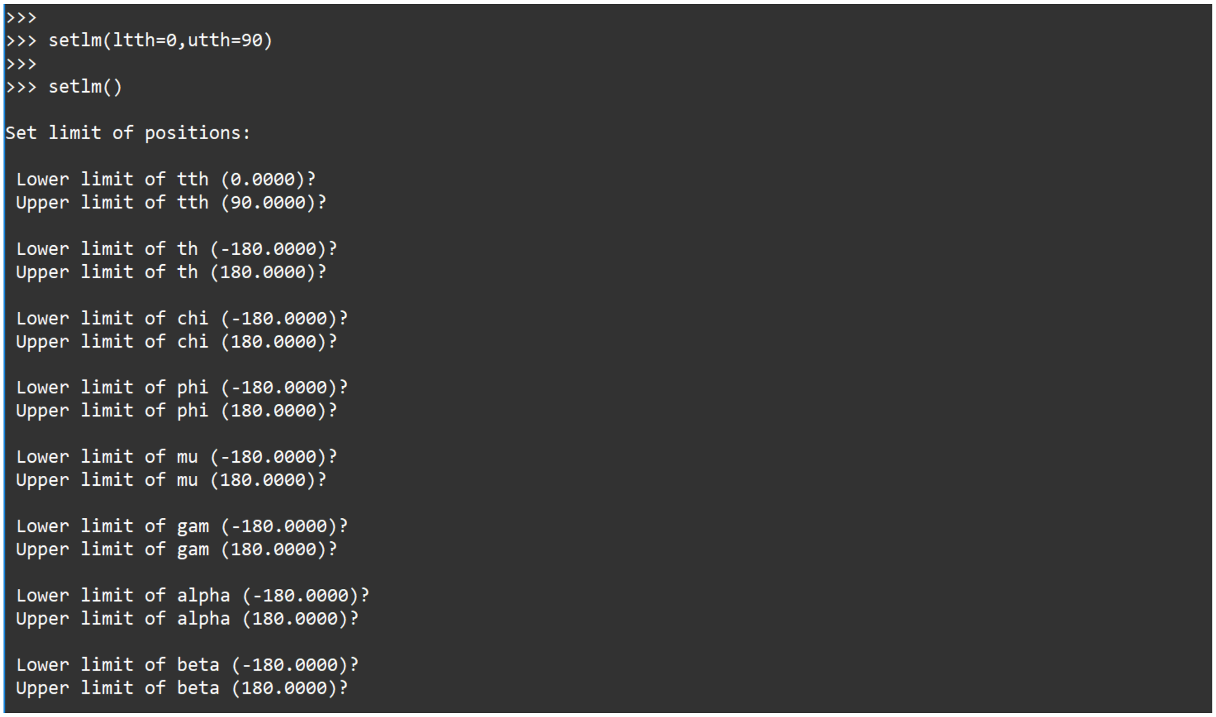


```
>>> freeze()

 Freeze mu (1.0000)?
 Freeze gam (2.0000)?
 Freeze omega (3.0000)?

Positions of frozen angles:
        mu       gam     omega
    1.0000    2.0000    3.0000
>>>
```

setlm()

setlm(ltth=?,utth=?,…)

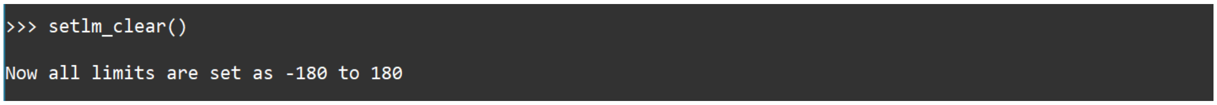
- Set limits of positions. This will be applied to selecting calculated positions.
- Valid keyword: l(u)tth, l(u)th, l(u)chi, l(u)phi, l(u)mu, l(u)gam, l(u)alpha, l(u)beta



```
>>>
>>> setlm(ltth=0,utth=90)
>>>
>>> setlm()

Set limit of positions:

 Lower limit of tth (0.0000)?
 Upper limit of tth (90.0000)?

 Lower limit of th (-180.0000)?
 Upper limit of th (180.0000)?

 Lower limit of chi (-180.0000)?
 Upper limit of chi (180.0000)?

 Lower limit of phi (-180.0000)?
 Upper limit of phi (180.0000)?

 Lower limit of mu (-180.0000)?
 Upper limit of mu (180.0000)?

 Lower limit of gam (-180.0000)?
 Upper limit of gam (180.0000)?

 Lower limit of alpha (-180.0000)?
 Upper limit of alpha (180.0000)?

 Lower limit of beta (-180.0000)?
 Upper limit of beta (180.0000)?
```

setlm_clear()

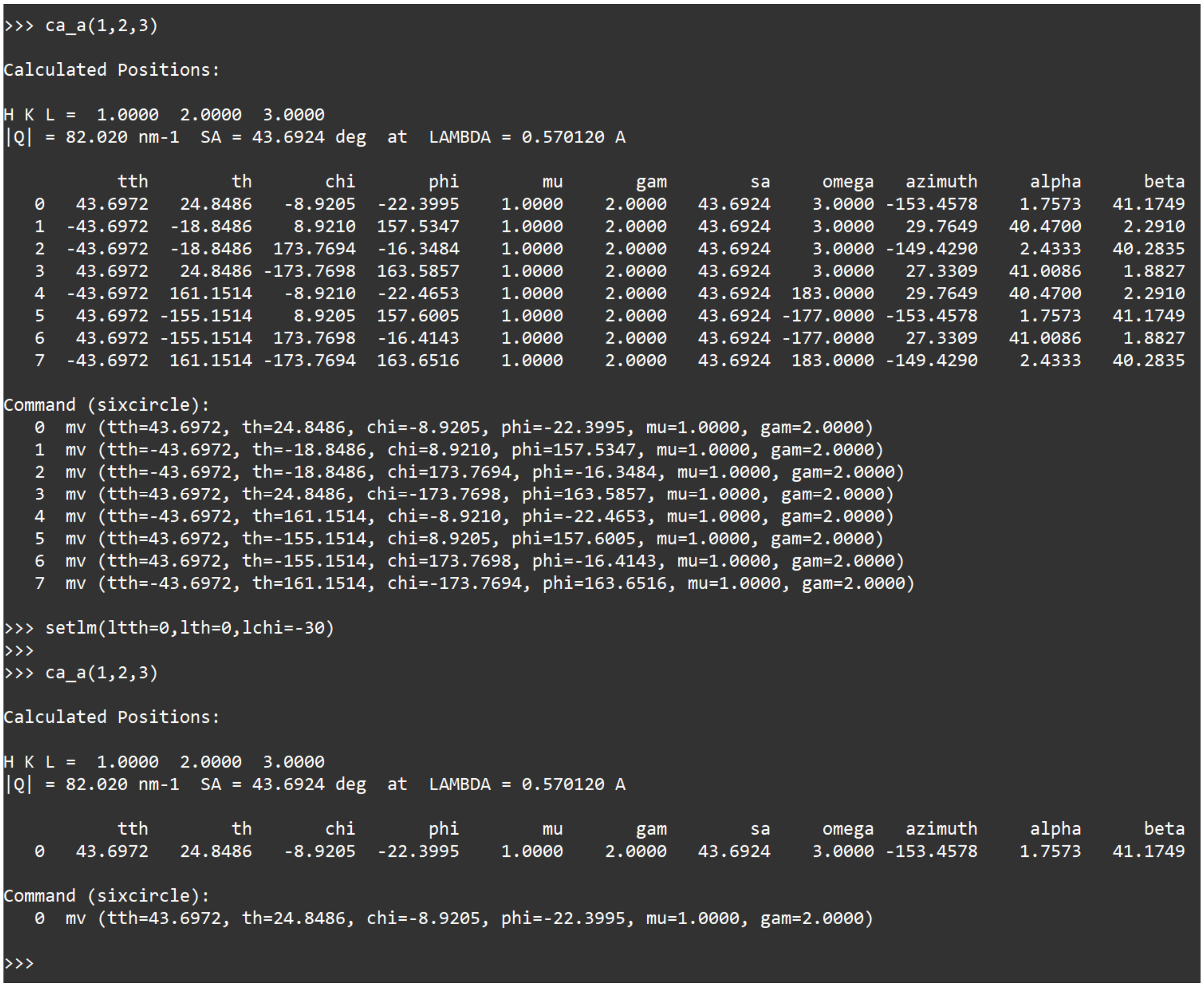
- Set all limits to -180 to 180



```
>>> setlm_clear()

Now all limits are set as -180 to 180
```

ca_a(H,K,L)

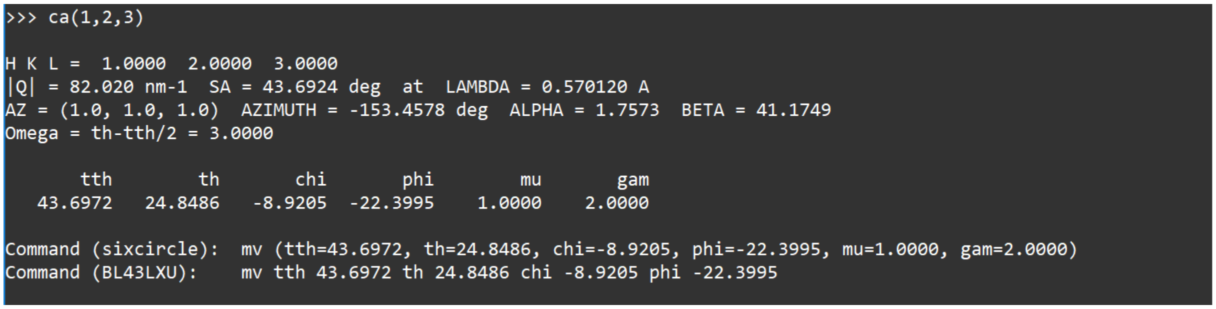
- Find all positions for H, K, L.
- All sets of positions within preset limits in preset mode are displayed.
- To select one set, use setlm.



```
>>> ca_a(1,2,3)

Calculated Positions:

H K L =  1.0000  2.0000  3.0000
|Q| = 82.020 nm-1  SA = 43.6924 deg  at  LAMBDA = 0.570120 A

           tth        th       chi       phi        mu       gam        sa     omega   azimuth     alpha      beta
   0   43.6972   24.8486   -8.9205  -22.3995    1.0000    2.0000   43.6924    3.0000 -153.4578    1.7573   41.1749
   1  -43.6972  -18.8486    8.9210  157.5347    1.0000    2.0000   43.6924    3.0000   29.7649   40.4700    2.2910
   2  -43.6972  -18.8486  173.7694  -16.3484    1.0000    2.0000   43.6924    3.0000 -149.4290    2.4333   40.2835
   3   43.6972   24.8486 -173.7698  163.5857    1.0000    2.0000   43.6924    3.0000   27.3309   41.0086    1.8827
   4  -43.6972  161.1514   -8.9210  -22.4653    1.0000    2.0000   43.6924  183.0000   29.7649   40.4700    2.2910
   5   43.6972 -155.1514    8.9205  157.6005    1.0000    2.0000   43.6924 -177.0000 -153.4578    1.7573   41.1749
   6   43.6972 -155.1514  173.7698  -16.4143    1.0000    2.0000   43.6924 -177.0000   27.3309   41.0086    1.8827
   7  -43.6972  161.1514 -173.7694  163.6516    1.0000    2.0000   43.6924  183.0000 -149.4290    2.4333   40.2835

Command (sixcircle):
   0  mv (tth=43.6972, th=24.8486, chi=-8.9205, phi=-22.3995, mu=1.0000, gam=2.0000)
   1  mv (tth=-43.6972, th=-18.8486, chi=8.9210, phi=157.5347, mu=1.0000, gam=2.0000)
   2  mv (tth=-43.6972, th=-18.8486, chi=173.7694, phi=-16.3484, mu=1.0000, gam=2.0000)
   3  mv (tth=43.6972, th=24.8486, chi=-173.7698, phi=163.5857, mu=1.0000, gam=2.0000)
   4  mv (tth=-43.6972, th=161.1514, chi=-8.9210, phi=-22.4653, mu=1.0000, gam=2.0000)
   5  mv (tth=43.6972, th=-155.1514, chi=8.9205, phi=157.6005, mu=1.0000, gam=2.0000)
   6  mv (tth=43.6972, th=-155.1514, chi=173.7698, phi=-16.4143, mu=1.0000, gam=2.0000)
   7  mv (tth=-43.6972, th=161.1514, chi=-173.7694, phi=163.6516, mu=1.0000, gam=2.0000)

>>> setlm(ltth=0,lth=0,lchi=-30)
>>>
>>> ca_a(1,2,3)

Calculated Positions:

H K L =  1.0000  2.0000  3.0000
|Q| = 82.020 nm-1  SA = 43.6924 deg  at  LAMBDA = 0.570120 A

           tth        th       chi       phi        mu       gam        sa     omega   azimuth     alpha      beta
   0   43.6972   24.8486   -8.9205  -22.3995    1.0000    2.0000   43.6924    3.0000 -153.4578    1.7573   41.1749

Command (sixcircle):
   0  mv (tth=43.6972, th=24.8486, chi=-8.9205, phi=-22.3995, mu=1.0000, gam=2.0000)

>>>
```

ca(H,K,L)

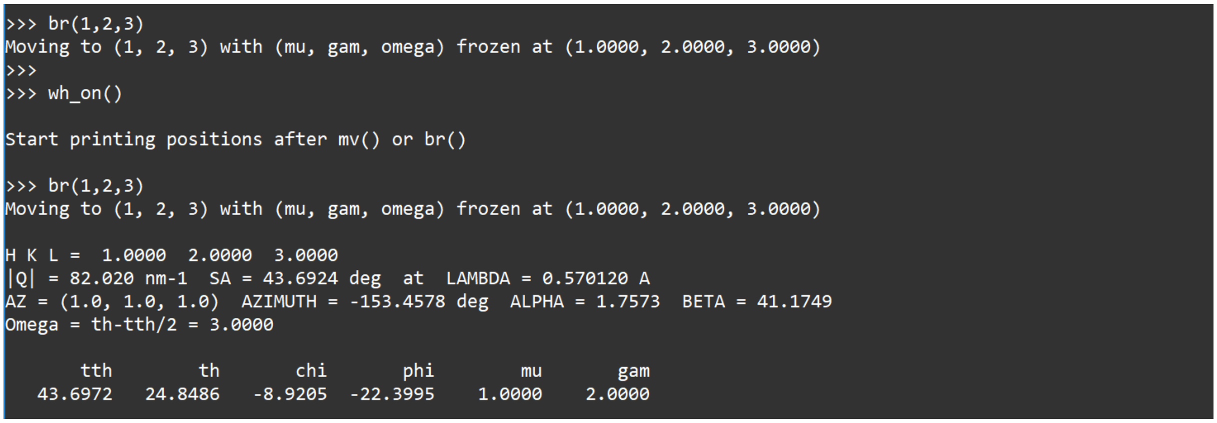
- Find positions for H, K, L.
- Only the first set of positions is displayed.



```
>>> ca(1,2,3)

H K L =  1.0000  2.0000  3.0000
|Q| = 82.020 nm-1  SA = 43.6924 deg  at  LAMBDA = 0.570120 A
AZ = (1.0, 1.0, 1.0)  AZIMUTH = -153.4578 deg  ALPHA = 1.7573  BETA = 41.1749
Omega = th-tth/2 = 3.0000

       tth        th       chi       phi        mu       gam
   43.6972   24.8486   -8.9205  -22.3995    1.0000    2.0000

Command (sixcircle):  mv (tth=43.6972, th=24.8486, chi=-8.9205, phi=-22.3995, mu=1.0000, gam=2.0000)
Command (BL43LXU):    mv tth 43.6972 th 24.8486 chi -8.9205 phi -22.3995
```

br(H,K,L)

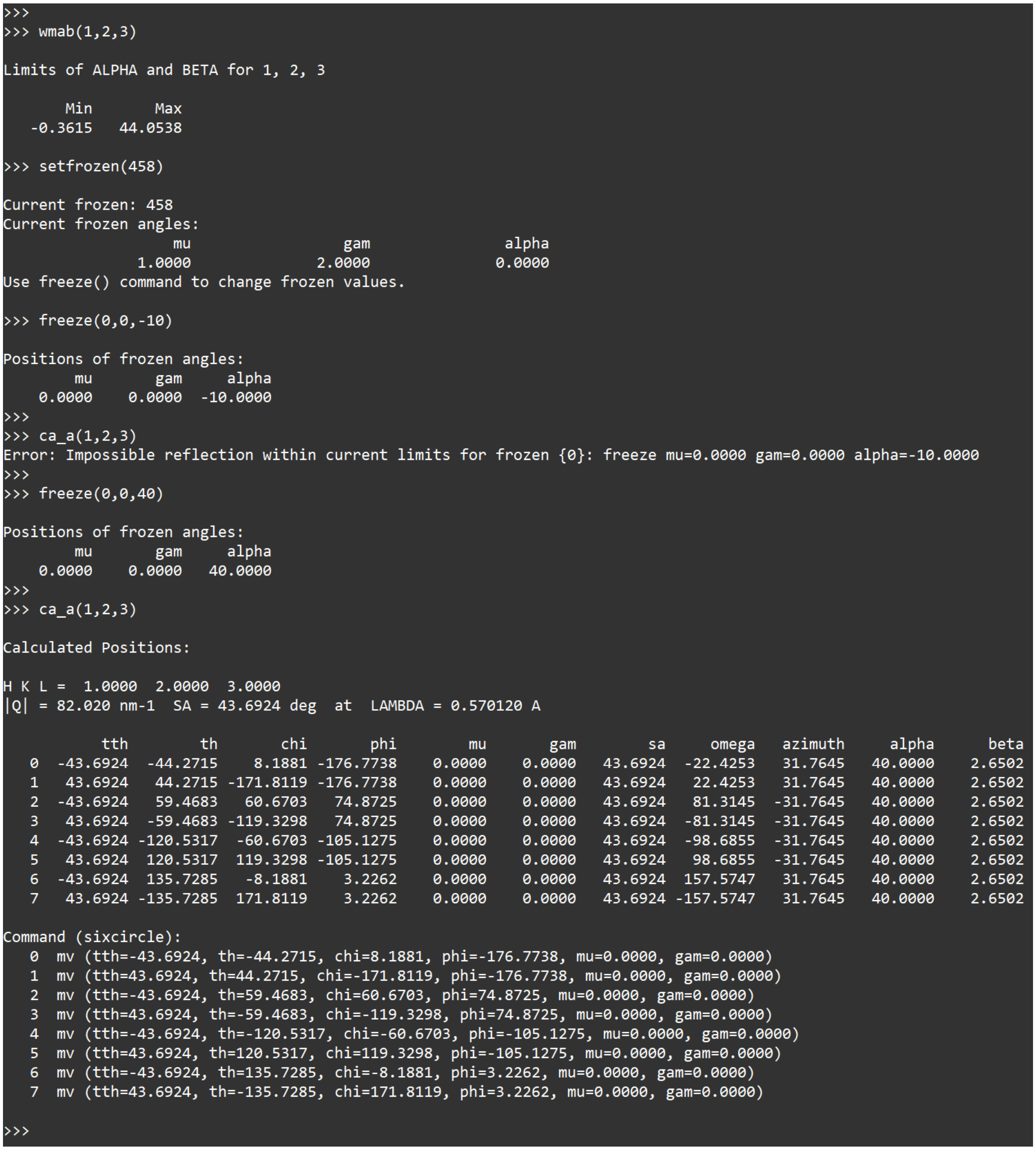
- Move to positions for H, K, L.
- The first set of positions is used.



```
>>> br(1,2,3)
Moving to (1, 2, 3) with (mu, gam, omega) frozen at (1.0000, 2.0000, 3.0000)
>>>
>>> wh_on()

Start printing positions after mv() or br()

>>> br(1,2,3)
Moving to (1, 2, 3) with (mu, gam, omega) frozen at (1.0000, 2.0000, 3.0000)

H K L =  1.0000  2.0000  3.0000
|Q| = 82.020 nm-1  SA = 43.6924 deg  at  LAMBDA = 0.570120 A
AZ = (1.0, 1.0, 1.0)  AZIMUTH = -153.4578 deg  ALPHA = 1.7573  BETA = 41.1749
Omega = th-tth/2 = 3.0000

       tth        th       chi       phi        mu       gam
   43.6972   24.8486   -8.9205  -22.3995    1.0000    2.0000
```

wmab(H, K, L)

- Check the limits of alpha and beta for H, K, L.
- This is simply determined by the deflection angle of (HKL) and the angle from (HKL) to the reference vector.

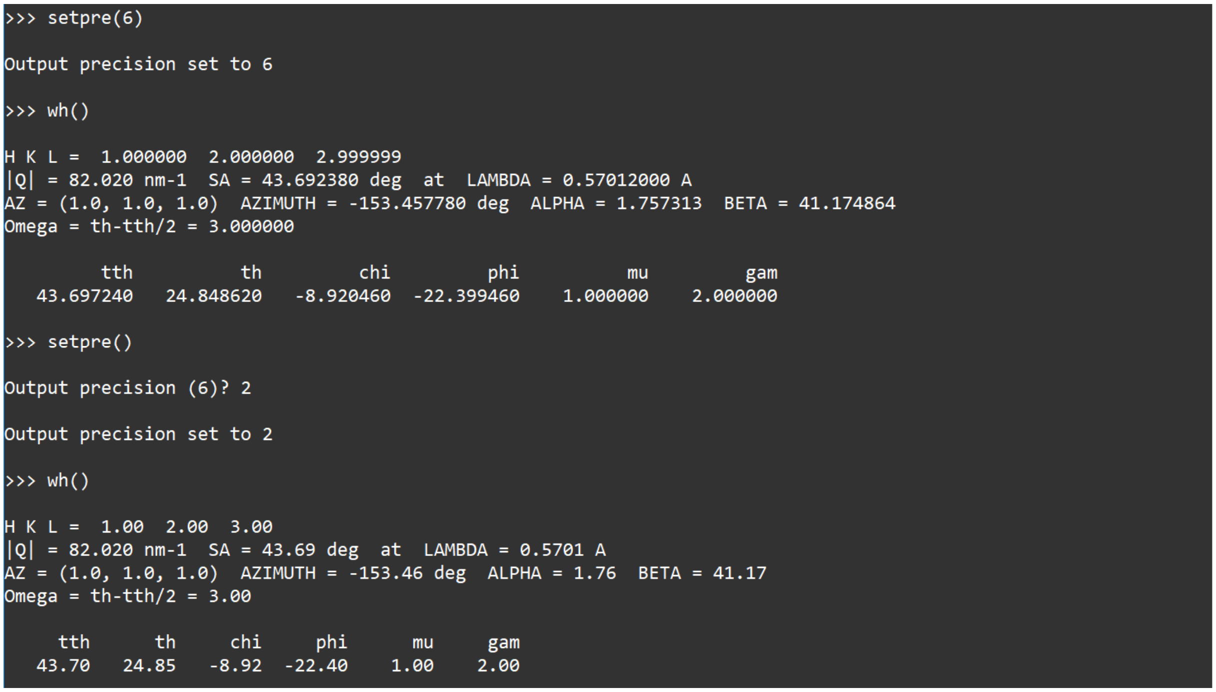


```
>>>
>>> wmab(1,2,3)

Limits of ALPHA and BETA for 1, 2, 3

       Min       Max
   -0.3615   44.0538

>>> setfrozen(458)

Current frozen: 458
Current frozen angles:
                  mu                  gam                alpha
              1.0000               2.0000               0.0000
Use freeze() command to change frozen values.

>>> freeze(0,0,-10)

Positions of frozen angles:
        mu       gam     alpha
    0.0000    0.0000  -10.0000
>>>
>>> ca_a(1,2,3)
Error: Impossible reflection within current limits for frozen {0}: freeze mu=0.0000 gam=0.0000 alpha=-10.0000
>>>
>>> freeze(0,0,40)

Positions of frozen angles:
        mu       gam     alpha
    0.0000    0.0000   40.0000
>>>
>>> ca_a(1,2,3)

Calculated Positions:

H K L =  1.0000  2.0000  3.0000
|Q| = 82.020 nm-1  SA = 43.6924 deg  at  LAMBDA = 0.570120 A

          tth        th       chi       phi        mu       gam        sa     omega   azimuth     alpha      beta
   0  -43.6924  -44.2715    8.1881 -176.7738    0.0000    0.0000   43.6924  -22.4253   31.7645   40.0000    2.6502
   1   43.6924   44.2715 -171.8119 -176.7738    0.0000    0.0000   43.6924   22.4253   31.7645   40.0000    2.6502
   2  -43.6924   59.4683   60.6703   74.8725    0.0000    0.0000   43.6924   81.3145  -31.7645   40.0000    2.6502
   3   43.6924  -59.4683 -119.3298   74.8725    0.0000    0.0000   43.6924  -81.3145  -31.7645   40.0000    2.6502
   4  -43.6924 -120.5317  -60.6703 -105.1275    0.0000    0.0000   43.6924  -98.6855  -31.7645   40.0000    2.6502
   5   43.6924  120.5317  119.3298 -105.1275    0.0000    0.0000   43.6924   98.6855  -31.7645   40.0000    2.6502
   6  -43.6924  135.7285   -8.1881    3.2262    0.0000    0.0000   43.6924  157.5747   31.7645   40.0000    2.6502
   7   43.6924 -135.7285  171.8119    3.2262    0.0000    0.0000   43.6924 -157.5747   31.7645   40.0000    2.6502

Command (sixcircle):
   0  mv (tth=-43.6924, th=-44.2715, chi=8.1881, phi=-176.7738, mu=0.0000, gam=0.0000)
   1  mv (tth=43.6924, th=44.2715, chi=-171.8119, phi=-176.7738, mu=0.0000, gam=0.0000)
   2  mv (tth=-43.6924, th=59.4683, chi=60.6703, phi=74.8725, mu=0.0000, gam=0.0000)
   3  mv (tth=43.6924, th=-59.4683, chi=-119.3298, phi=74.8725, mu=0.0000, gam=0.0000)
   4  mv (tth=-43.6924, th=-120.5317, chi=-60.6703, phi=-105.1275, mu=0.0000, gam=0.0000)
   5  mv (tth=43.6924, th=120.5317, chi=119.3298, phi=-105.1275, mu=0.0000, gam=0.0000)
   6  mv (tth=-43.6924, th=135.7285, chi=-8.1881, phi=3.2262, mu=0.0000, gam=0.0000)
   7  mv (tth=43.6924, th=-135.7285, chi=171.8119, phi=3.2262, mu=0.0000, gam=0.0000)

>>>
```

setpre()

setpre(4)

- Set the number of digits in output.



```
>>> setpre(6)

Output precision set to 6

>>> wh()

H K L =  1.000000  2.000000  2.999999
|Q| = 82.020 nm-1  SA = 43.692380 deg  at  LAMBDA = 0.57012000 A
AZ = (1.0, 1.0, 1.0)  AZIMUTH = -153.457780 deg  ALPHA = 1.757313  BETA = 41.174864
Omega = th-tth/2 = 3.000000

         tth          th         chi         phi          mu         gam
   43.697240   24.848620   -8.920460  -22.399460    1.000000    2.000000

>>> setpre()

Output precision (6)? 2

Output precision set to 2

>>> wh()

H K L =  1.00  2.00  3.00
|Q| = 82.020 nm-1  SA = 43.69 deg  at  LAMBDA = 0.5701 A
AZ = (1.0, 1.0, 1.0)  AZIMUTH = -153.46 deg  ALPHA = 1.76  BETA = 41.17
Omega = th-tth/2 = 3.00

     tth      th     chi     phi      mu     gam
   43.70   24.85   -8.92  -22.40    1.00    2.00
```

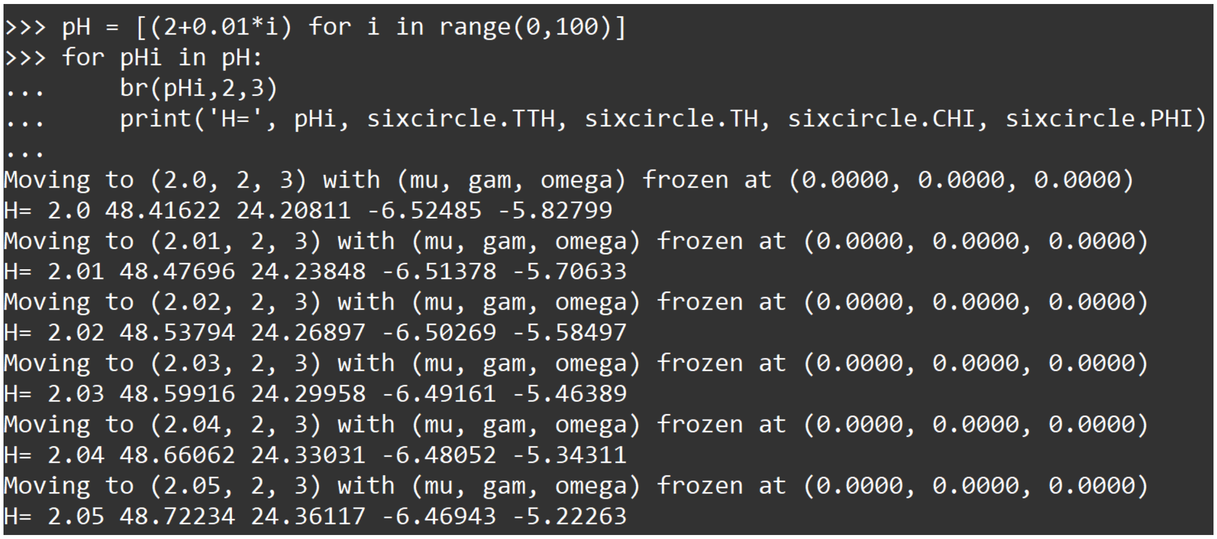
Discussion on multiple sets of positions

- There are numerous sets of positions for one HKL vector. When freezing three angles, the number of sets becomes finite, generally 2, 4, 6, 8, or more, depending on frozen angles and corresponding positions. Please note these sets are different from the eight sectors in FOURC (https://certif.com/spec_manual/fourc_4_6.html), and, when mu or gamma is nonzero, the eight sectors in FOURC are invalid in SIXC geometry.

- Sets of positions are arranged in a simple order: Firstly compare the distance from tth to zero, then compare the th, then chi, phi, mu, gamma. It is necessary to arrange them in a better order.

Loops and branches in python shell

- Users can run some simple loops and branches directly in python shell.
- For example:



```
>>> pH = [(2+0.01*i) for i in range(0,100)]
>>> for pHi in pH:
...     br(pHi,2,3)
...     print('H=', pHi, sixcircle.TTH, sixcircle.TH, sixcircle.CHI, sixcircle.PHI)
...
Moving to (2.0, 2, 3) with (mu, gam, omega) frozen at (0.0000, 0.0000, 0.0000)
H= 2.0 48.41622 24.20811 -6.52485 -5.82799
Moving to (2.01, 2, 3) with (mu, gam, omega) frozen at (0.0000, 0.0000, 0.0000)
H= 2.01 48.47696 24.23848 -6.51378 -5.70633
Moving to (2.02, 2, 3) with (mu, gam, omega) frozen at (0.0000, 0.0000, 0.0000)
H= 2.02 48.53794 24.26897 -6.50269 -5.58497
Moving to (2.03, 2, 3) with (mu, gam, omega) frozen at (0.0000, 0.0000, 0.0000)
H= 2.03 48.59916 24.29958 -6.49161 -5.46389
Moving to (2.04, 2, 3) with (mu, gam, omega) frozen at (0.0000, 0.0000, 0.0000)
H= 2.04 48.66062 24.33031 -6.48052 -5.34311
Moving to (2.05, 2, 3) with (mu, gam, omega) frozen at (0.0000, 0.0000, 0.0000)
H= 2.05 48.72234 24.36117 -6.46943 -5.22263
```

To prepare user’s own toolkit

- Users can also prepare their own toolkit using sixcircle.py.

- One example is **sixcircle_rqd.py**

- Put the toolkit.py in the same directory with sixc.py.
- import sixcircle; from sixcircle import *, so that it can visit functions and global variables in sixcircle.py.
- After preparing the toolkit, enter from toolkit import * in python shell after starting sixcircle.

List of global variables:

# Current positions
TTH, TH, OMEGA, CHI, PHI, OMEGA, MU, GAM, SA, ABSQ

# Current H, K, L, alpha, beta, azimuth
H, K, L, ALPHA, BETA, AZIMUTH

# frozen mode
g_frozen

# Frozen positions of angles:
F_TTH, F_TH, F_CHI, F_PHI, F_MU, F_GAM, F_AZIMUTH, F_OMEGA, F_ALPHA, F_BETA

# Sample description
g_sample

# Azimuth reference vector H, K, L
g_haz, g_kaz, g_laz

# Lattice parameters: a, b, c, alpha, beta, gamma
g_aa, g_bb, g_cc, g_al, g_be, g_ga

# Primary reflection: H, K, L, wavelength
g_h0, g_k0, g_l0, g_lambda0

To prepare user's own toolkit

# Primary reflection: positions of tth, th, chi, phi, mu, gam
g_u00, g_u01, g_u02, g_u03, g_u04, g_u05

# Secondary reflections: H, K, L
g_h1, g_k1, g_l1, g_lambda1

# Secondary reflection: positions of tth, th, chi, phi, mu, gam
g_u10, g_u11, g_u12, g_u13, g_u14, g_u15

# Wavelength in current calculation
LAMBDA

# Output precision
PRE

# Flag of wa_on or wa_off
FLAG_WA

- To visit one global variable, for example, sixcircle.g_frozen, other than ~~g_frozen.~~

- It is not recommended to modify global variables directly.

Two additional functions are also available in preparing toolkit:

flag, pos = ca_s(H, K, L)

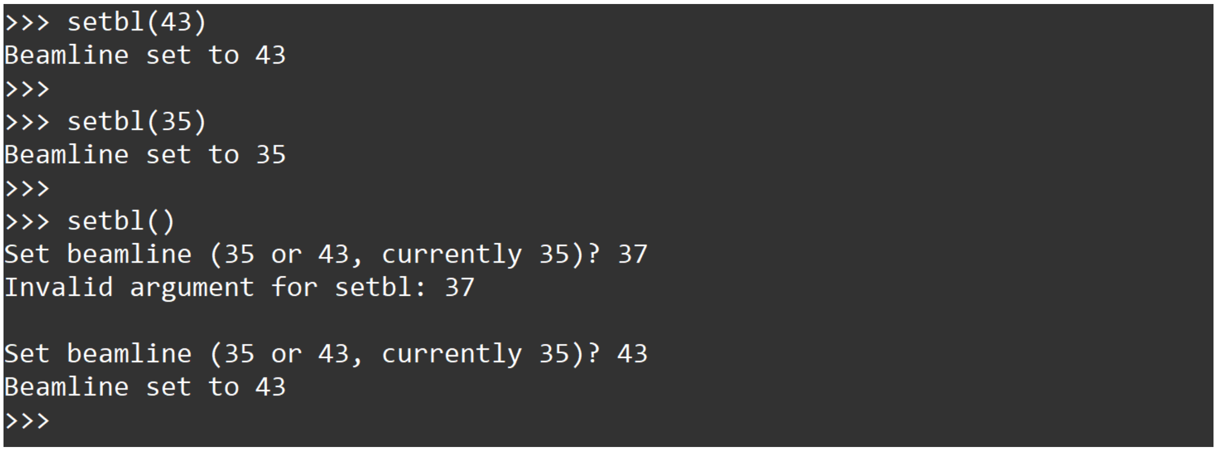
- Calculate positions in specified mode
- '_s': silent, calculation in background
- flag: True (False) when calculation is successful (failed: no solution)
- pos: N set of positions, [[mu1, gam1, tth1, th1, chi1, phi1, sa1, azimuth1, alpha1, beta1, omega1],[mu2,...],...,[muN,...]]

flag, min, max = wmab_s(H, K, L)

Check where the limits of alpha and beta are for H, K, L
'_s': silent, calculation in background
flag: True (False) when calculation is successful (failed: impossible HKL reflection)

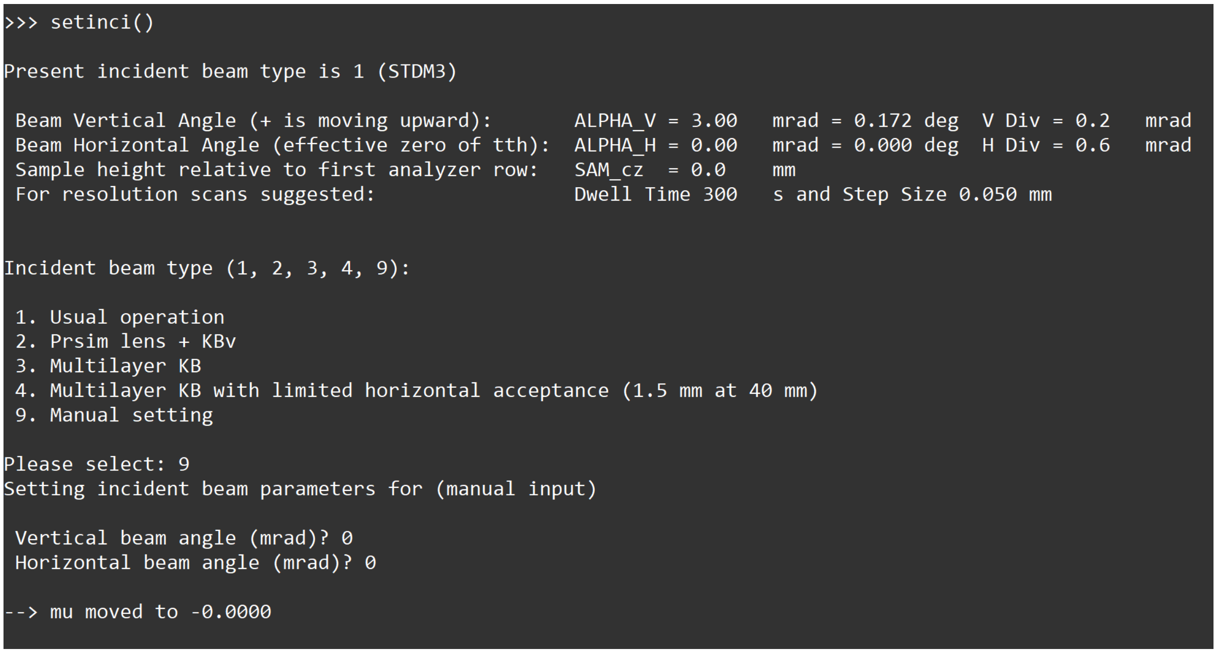
About sixcircle_rqd

setbl()
setbl(43) or setbl(35)



```
>>> setbl(43)
Beamline set to 43
>>>
>>> setbl(35)
Beamline set to 35
>>>
>>> setbl()
Set beamline (35 or 43, currently 35)? 37
Invalid argument for setbl: 37

Set beamline (35 or 43, currently 35)? 43
Beamline set to 43
>>>
```

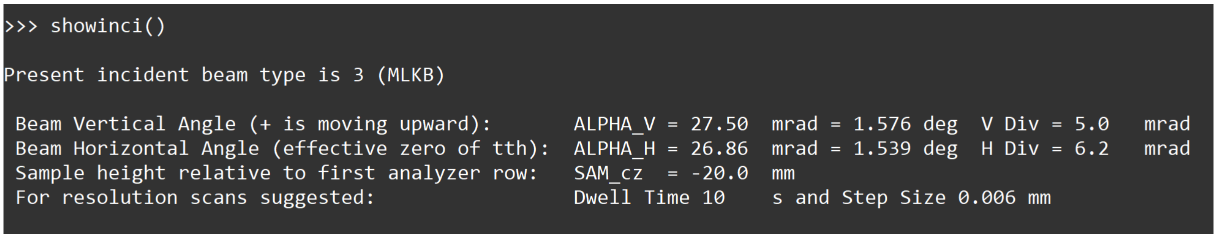
setinci()
setinci(1)



```
>>> setinci()

Present incident beam type is 1 (STDM3)

 Beam Vertical Angle (+ is moving upward):       ALPHA_V = 3.00   mrad = 0.172 deg  V Div = 0.2   mrad
 Beam Horizontal Angle (effective zero of tth):  ALPHA_H = 0.00   mrad = 0.000 deg  H Div = 0.6   mrad
 Sample height relative to first analyzer row:   SAM_cz  = 0.0    mm
 For resolution scans suggested:                 Dwell Time 300   s and Step Size 0.050 mm


Incident beam type (1, 2, 3, 4, 9):

 1. Usual operation
 2. Prsim lens + KBv
 3. Multilayer KB
 4. Multilayer KB with limited horizontal acceptance (1.5 mm at 40 mm)
 9. Manual setting

Please select: 9
Setting incident beam parameters for (manual input)

 Vertical beam angle (mrad)? 0
 Horizontal beam angle (mrad)? 0

--> mu moved to -0.0000
```

About sixcircle_rqd

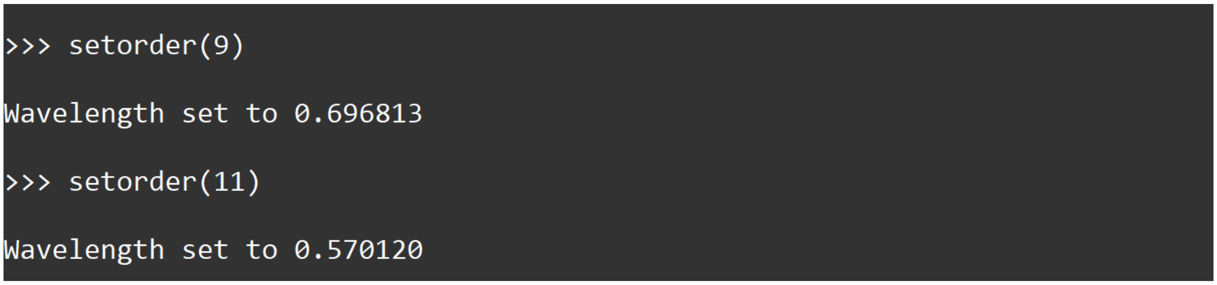
showinci()



```
>>> showinci()

Present incident beam type is 3 (MLKB)

 Beam Vertical Angle (+ is moving upward):       ALPHA_V = 27.50  mrad = 1.576 deg  V Div = 5.0   mrad
 Beam Horizontal Angle (effective zero of tth):  ALPHA_H = 26.86  mrad = 1.539 deg  H Div = 6.2   mrad
 Sample height relative to first analyzer row:   SAM_cz  = -20.0  mm
 For resolution scans suggested:                 Dwell Time 10    s and Step Size 0.006 mm
```

setorder()

setorder(11)



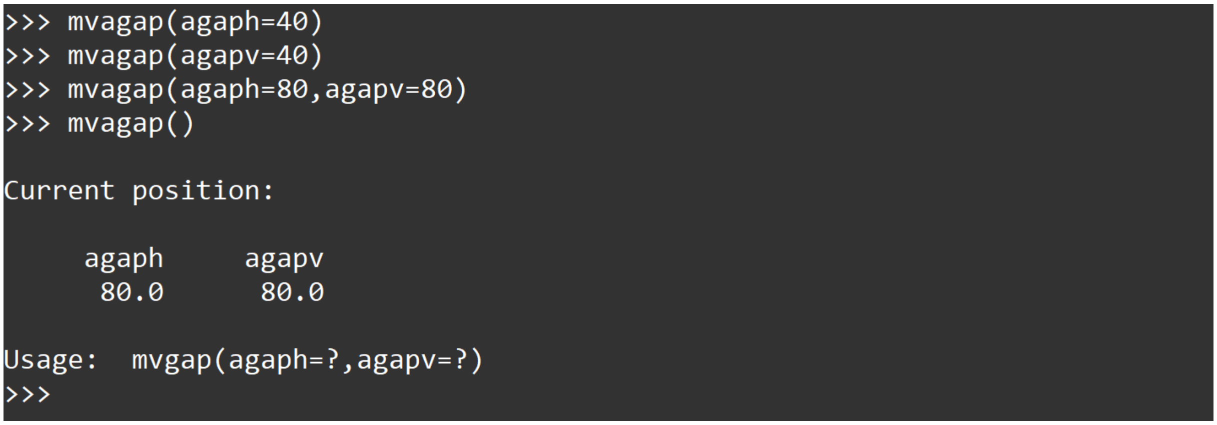
```
>>> setorder(9)

Wavelength set to 0.696813

>>> setorder(11)

Wavelength set to 0.570120
```

mvagap()

- Check agaph, agapv

mvagap(agaph=?, agapv=?)

- Move agaph, agapv



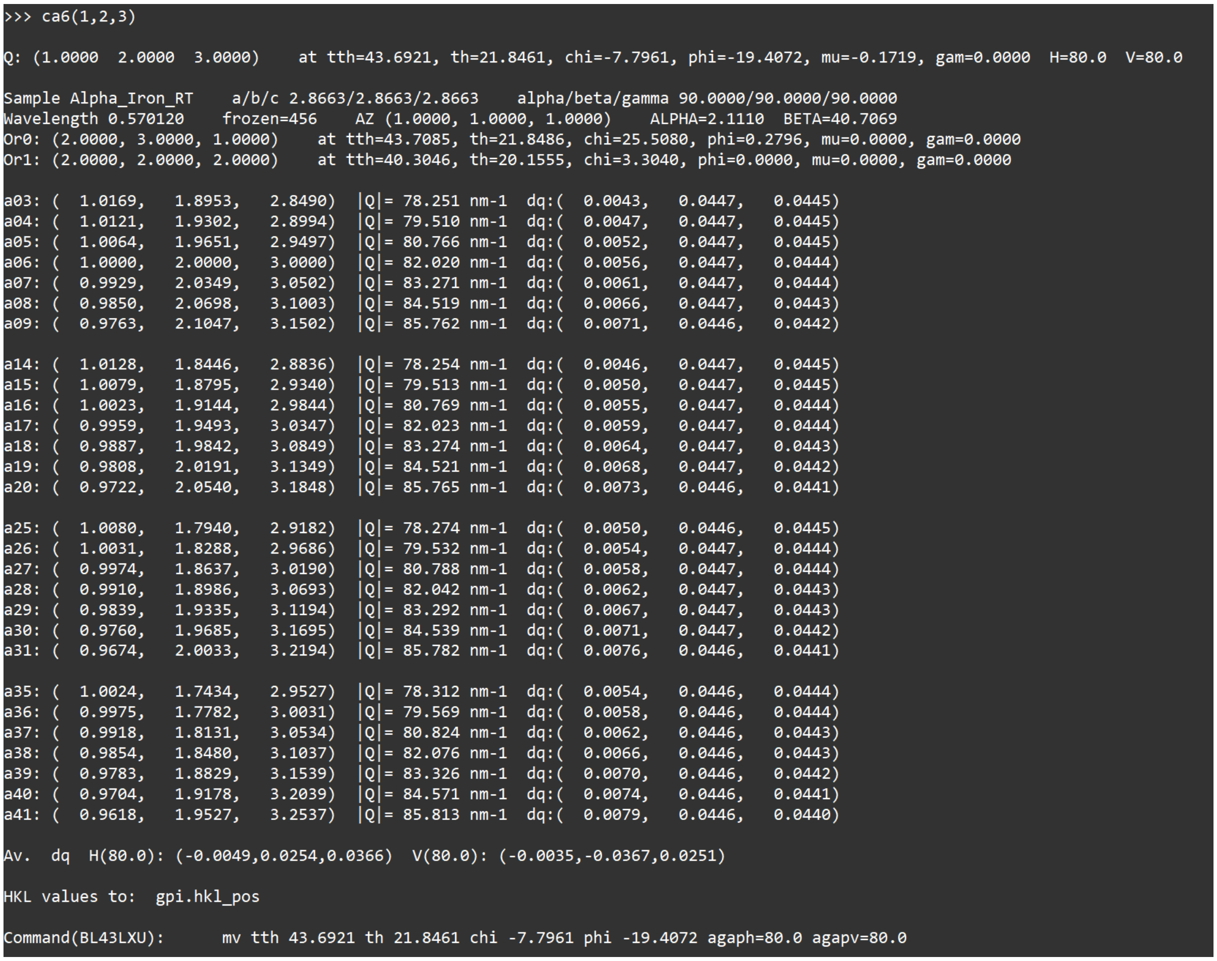
```
>>> mvagap(agaph=40)
>>> mvagap(agapv=40)
>>> mvagap(agaph=80,agapv=80)
>>> mvagap()

Current position:

     agaph     agapv
      80.0      80.0

Usage:  mvgap(agaph=?,agapv=?)
>>>
```

ca6(H, K, L)

ca6(H, K, L, tH, tK, tL)

- Calculate reflection of every analyzer when A06 is put to H, K, L.
- Optional vector tau=(tH, tK, tL) represents the reference reciprocal lattice point.
- Do calculation for the first set of positions. File “gpi.hkl_pos” will be created.

For bl43lxu:



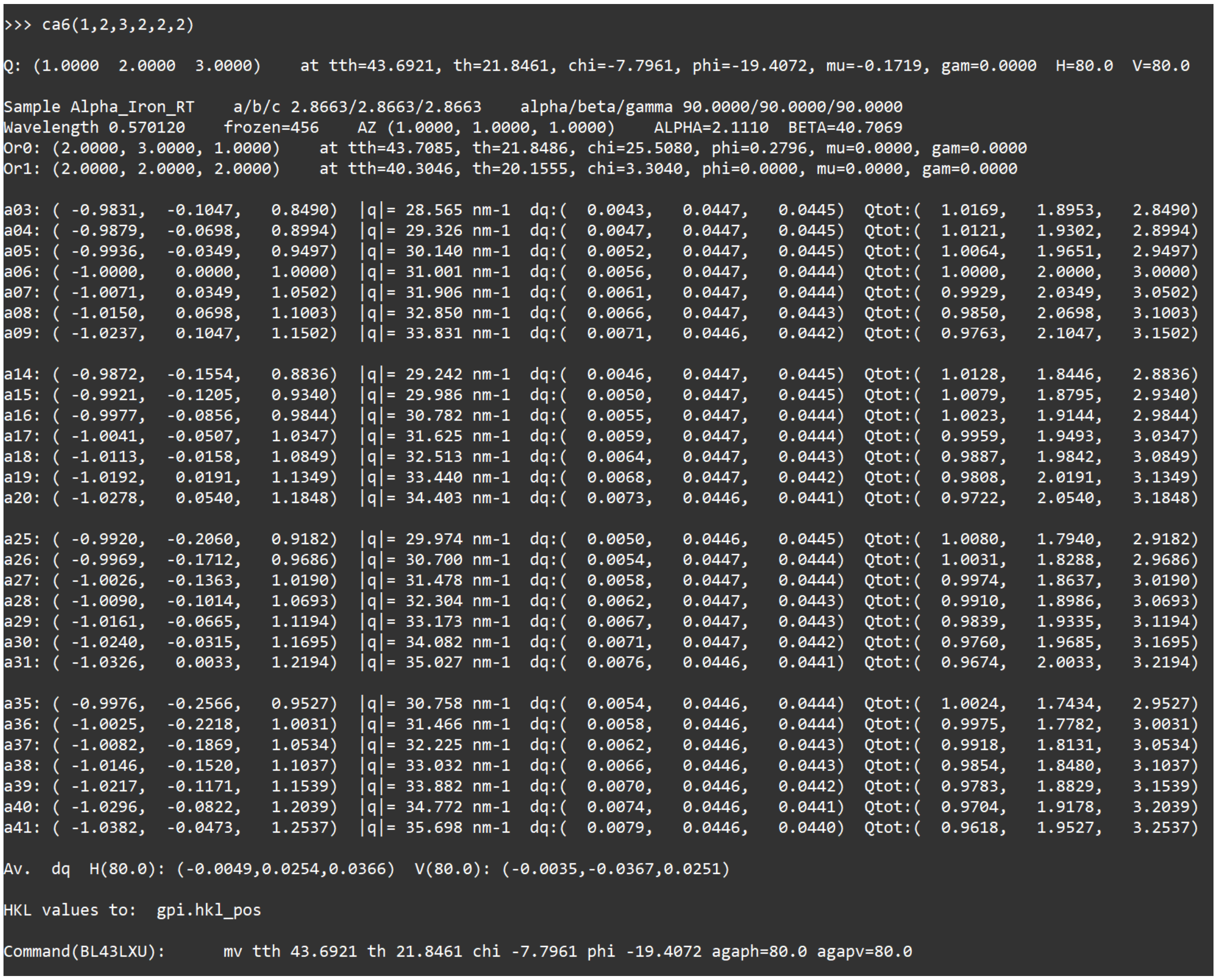
```
>>> ca6(1,2,3)

Q: (1.0000  2.0000  3.0000)    at tth=43.6921, th=21.8461, chi=-7.7961, phi=-19.4072, mu=-0.1719, gam=0.0000  H=80.0  V=80.0

Sample Alpha_Iron_RT    a/b/c 2.8663/2.8663/2.8663    alpha/beta/gamma 90.0000/90.0000/90.0000
Wavelength 0.570120    frozen=456    AZ (1.0000, 1.0000, 1.0000)    ALPHA=2.1110  BETA=40.7069
Or0: (2.0000, 3.0000, 1.0000)    at tth=43.7085, th=21.8486, chi=25.5080, phi=0.2796, mu=0.0000, gam=0.0000
Or1: (2.0000, 2.0000, 2.0000)    at tth=40.3046, th=20.1555, chi=3.3040, phi=0.0000, mu=0.0000, gam=0.0000

a03: (  1.0169,   1.8953,   2.8490)  |Q|= 78.251 nm-1  dq:(  0.0043,   0.0447,   0.0445)
a04: (  1.0121,   1.9302,   2.8994)  |Q|= 79.510 nm-1  dq:(  0.0047,   0.0447,   0.0445)
a05: (  1.0064,   1.9651,   2.9497)  |Q|= 80.766 nm-1  dq:(  0.0052,   0.0447,   0.0445)
a06: (  1.0000,   2.0000,   3.0000)  |Q|= 82.020 nm-1  dq:(  0.0056,   0.0447,   0.0444)
a07: (  0.9929,   2.0349,   3.0502)  |Q|= 83.271 nm-1  dq:(  0.0061,   0.0447,   0.0444)
a08: (  0.9850,   2.0698,   3.1003)  |Q|= 84.519 nm-1  dq:(  0.0066,   0.0447,   0.0443)
a09: (  0.9763,   2.1047,   3.1502)  |Q|= 85.762 nm-1  dq:(  0.0071,   0.0446,   0.0442)

a14: (  1.0128,   1.8446,   2.8836)  |Q|= 78.254 nm-1  dq:(  0.0046,   0.0447,   0.0445)
a15: (  1.0079,   1.8795,   2.9340)  |Q|= 79.513 nm-1  dq:(  0.0050,   0.0447,   0.0445)
a16: (  1.0023,   1.9144,   2.9844)  |Q|= 80.769 nm-1  dq:(  0.0055,   0.0447,   0.0444)
a17: (  0.9959,   1.9493,   3.0347)  |Q|= 82.023 nm-1  dq:(  0.0059,   0.0447,   0.0444)
a18: (  0.9887,   1.9842,   3.0849)  |Q|= 83.274 nm-1  dq:(  0.0064,   0.0447,   0.0443)
a19: (  0.9808,   2.0191,   3.1349)  |Q|= 84.521 nm-1  dq:(  0.0068,   0.0447,   0.0442)
a20: (  0.9722,   2.0540,   3.1848)  |Q|= 85.765 nm-1  dq:(  0.0073,   0.0446,   0.0441)

a25: (  1.0080,   1.7940,   2.9182)  |Q|= 78.274 nm-1  dq:(  0.0050,   0.0446,   0.0445)
a26: (  1.0031,   1.8288,   2.9686)  |Q|= 79.532 nm-1  dq:(  0.0054,   0.0447,   0.0444)
a27: (  0.9974,   1.8637,   3.0190)  |Q|= 80.788 nm-1  dq:(  0.0058,   0.0447,   0.0444)
a28: (  0.9910,   1.8986,   3.0693)  |Q|= 82.042 nm-1  dq:(  0.0062,   0.0447,   0.0443)
a29: (  0.9839,   1.9335,   3.1194)  |Q|= 83.292 nm-1  dq:(  0.0067,   0.0447,   0.0443)
a30: (  0.9760,   1.9685,   3.1695)  |Q|= 84.539 nm-1  dq:(  0.0071,   0.0447,   0.0442)
a31: (  0.9674,   2.0033,   3.2194)  |Q|= 85.782 nm-1  dq:(  0.0076,   0.0446,   0.0441)

a35: (  1.0024,   1.7434,   2.9527)  |Q|= 78.312 nm-1  dq:(  0.0054,   0.0446,   0.0444)
a36: (  0.9975,   1.7782,   3.0031)  |Q|= 79.569 nm-1  dq:(  0.0058,   0.0446,   0.0444)
a37: (  0.9918,   1.8131,   3.0534)  |Q|= 80.824 nm-1  dq:(  0.0062,   0.0446,   0.0443)
a38: (  0.9854,   1.8480,   3.1037)  |Q|= 82.076 nm-1  dq:(  0.0066,   0.0446,   0.0443)
a39: (  0.9783,   1.8829,   3.1539)  |Q|= 83.326 nm-1  dq:(  0.0070,   0.0446,   0.0442)
a40: (  0.9704,   1.9178,   3.2039)  |Q|= 84.571 nm-1  dq:(  0.0074,   0.0446,   0.0441)
a41: (  0.9618,   1.9527,   3.2537)  |Q|= 85.813 nm-1  dq:(  0.0079,   0.0446,   0.0440)

Av.  dq  H(80.0): (-0.0049,0.0254,0.0366)  V(80.0): (-0.0035,-0.0367,0.0251)

HKL values to:  gpi.hkl_pos

Command(BL43LXU):      mv tth 43.6921 th 21.8461 chi -7.7961 phi -19.4072 agaph=80.0 agapv=80.0
```

ca6(H, K, L)

ca6(H, K, L, tH, tK, tL)

- Calculate reflection of every analyzer when A06 is put to H, K, L.
- Optional vector tau=(tH, tK, tL) represents the reference reciprocal lattice point.
- Do calculation for the first set of positions. File “gpi.hkl_pos” will be created.

For bl43lxu:

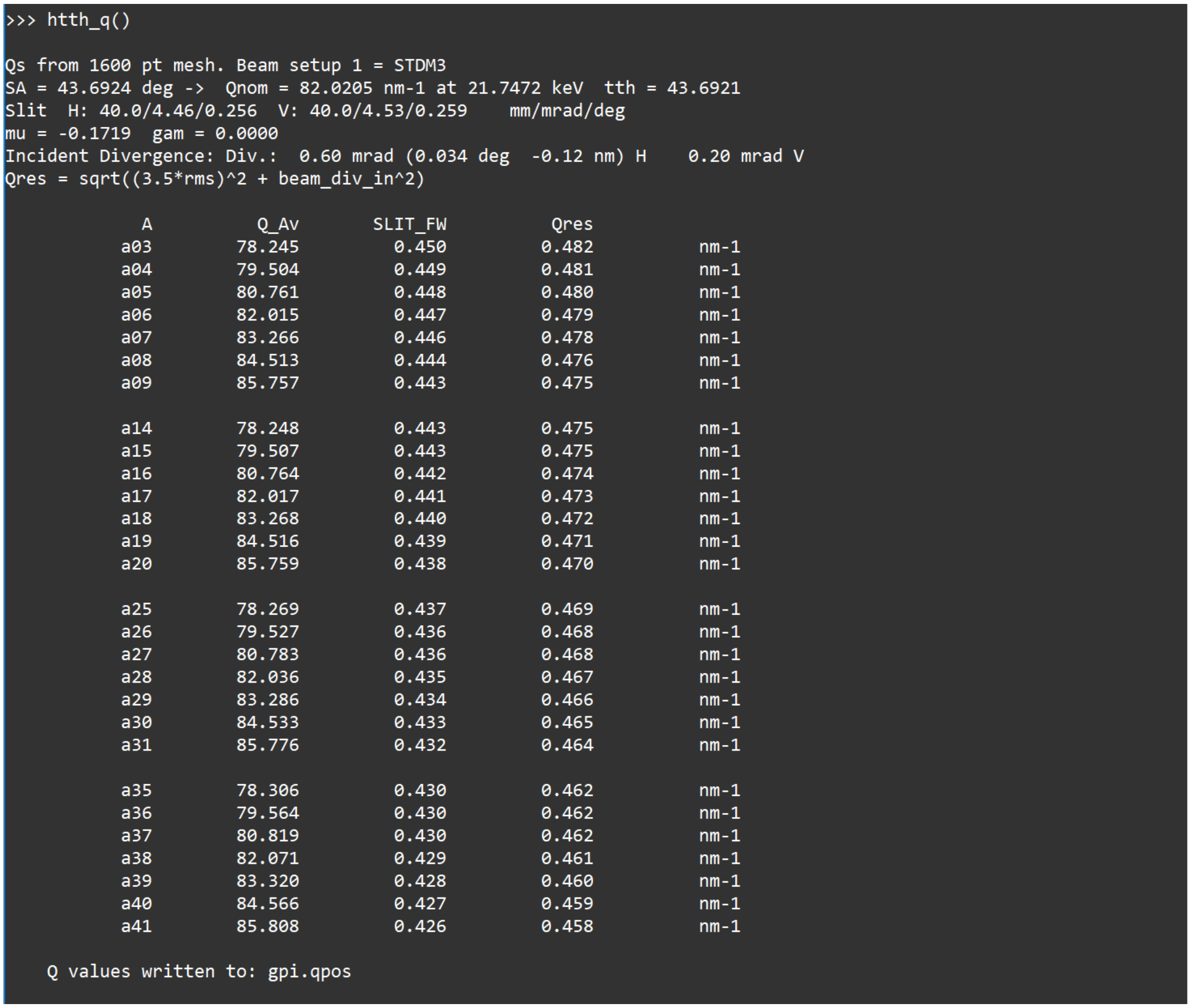


```
>>> ca6(1,2,3,2,2,2)

Q: (1.0000  2.0000  3.0000)    at tth=43.6921, th=21.8461, chi=-7.7961, phi=-19.4072, mu=-0.1719, gam=0.0000  H=80.0  V=80.0

Sample Alpha_Iron_RT    a/b/c 2.8663/2.8663/2.8663    alpha/beta/gamma 90.0000/90.0000/90.0000
Wavelength 0.570120    frozen=456    AZ (1.0000, 1.0000, 1.0000)    ALPHA=2.1110  BETA=40.7069
Or0: (2.0000, 3.0000, 1.0000)    at tth=43.7085, th=21.8486, chi=25.5080, phi=0.2796, mu=0.0000, gam=0.0000
Or1: (2.0000, 2.0000, 2.0000)    at tth=40.3046, th=20.1555, chi=3.3040, phi=0.0000, mu=0.0000, gam=0.0000

a03: ( -0.9831,  -0.1047,   0.8490)  |q|= 28.565 nm-1  dq:(  0.0043,   0.0447,   0.0445)  Qtot:(  1.0169,   1.8953,   2.8490)
a04: ( -0.9879,  -0.0698,   0.8994)  |q|= 29.326 nm-1  dq:(  0.0047,   0.0447,   0.0445)  Qtot:(  1.0121,   1.9302,   2.8994)
a05: ( -0.9936,  -0.0349,   0.9497)  |q|= 30.140 nm-1  dq:(  0.0052,   0.0447,   0.0445)  Qtot:(  1.0064,   1.9651,   2.9497)
a06: ( -1.0000,   0.0000,   1.0000)  |q|= 31.001 nm-1  dq:(  0.0056,   0.0447,   0.0444)  Qtot:(  1.0000,   2.0000,   3.0000)
a07: ( -1.0071,   0.0349,   1.0502)  |q|= 31.906 nm-1  dq:(  0.0061,   0.0447,   0.0444)  Qtot:(  0.9929,   2.0349,   3.0502)
a08: ( -1.0150,   0.0698,   1.1003)  |q|= 32.850 nm-1  dq:(  0.0066,   0.0447,   0.0443)  Qtot:(  0.9850,   2.0698,   3.1003)
a09: ( -1.0237,   0.1047,   1.1502)  |q|= 33.831 nm-1  dq:(  0.0071,   0.0446,   0.0442)  Qtot:(  0.9763,   2.1047,   3.1502)

a14: ( -0.9872,  -0.1554,   0.8836)  |q|= 29.242 nm-1  dq:(  0.0046,   0.0447,   0.0445)  Qtot:(  1.0128,   1.8446,   2.8836)
a15: ( -0.9921,  -0.1205,   0.9340)  |q|= 29.986 nm-1  dq:(  0.0050,   0.0447,   0.0445)  Qtot:(  1.0079,   1.8795,   2.9340)
a16: ( -0.9977,  -0.0856,   0.9844)  |q|= 30.782 nm-1  dq:(  0.0055,   0.0447,   0.0444)  Qtot:(  1.0023,   1.9144,   2.9844)
a17: ( -1.0041,  -0.0507,   1.0347)  |q|= 31.625 nm-1  dq:(  0.0059,   0.0447,   0.0444)  Qtot:(  0.9959,   1.9493,   3.0347)
a18: ( -1.0113,  -0.0158,   1.0849)  |q|= 32.513 nm-1  dq:(  0.0064,   0.0447,   0.0443)  Qtot:(  0.9887,   1.9842,   3.0849)
a19: ( -1.0192,   0.0191,   1.1349)  |q|= 33.440 nm-1  dq:(  0.0068,   0.0447,   0.0442)  Qtot:(  0.9808,   2.0191,   3.1349)
a20: ( -1.0278,   0.0540,   1.1848)  |q|= 34.403 nm-1  dq:(  0.0073,   0.0446,   0.0441)  Qtot:(  0.9722,   2.0540,   3.1848)

a25: ( -0.9920,  -0.2060,   0.9182)  |q|= 29.974 nm-1  dq:(  0.0050,   0.0446,   0.0445)  Qtot:(  1.0080,   1.7940,   2.9182)
a26: ( -0.9969,  -0.1712,   0.9686)  |q|= 30.700 nm-1  dq:(  0.0054,   0.0447,   0.0444)  Qtot:(  1.0031,   1.8288,   2.9686)
a27: ( -1.0026,  -0.1363,   1.0190)  |q|= 31.478 nm-1  dq:(  0.0058,   0.0447,   0.0444)  Qtot:(  0.9974,   1.8637,   3.0190)
a28: ( -1.0090,  -0.1014,   1.0693)  |q|= 32.304 nm-1  dq:(  0.0062,   0.0447,   0.0443)  Qtot:(  0.9910,   1.8986,   3.0693)
a29: ( -1.0161,  -0.0665,   1.1194)  |q|= 33.173 nm-1  dq:(  0.0067,   0.0447,   0.0443)  Qtot:(  0.9839,   1.9335,   3.1194)
a30: ( -1.0240,  -0.0315,   1.1695)  |q|= 34.082 nm-1  dq:(  0.0071,   0.0447,   0.0442)  Qtot:(  0.9760,   1.9685,   3.1695)
a31: ( -1.0326,   0.0033,   1.2194)  |q|= 35.027 nm-1  dq:(  0.0076,   0.0446,   0.0441)  Qtot:(  0.9674,   2.0033,   3.2194)

a35: ( -0.9976,  -0.2566,   0.9527)  |q|= 30.758 nm-1  dq:(  0.0054,   0.0446,   0.0444)  Qtot:(  1.0024,   1.7434,   2.9527)
a36: ( -1.0025,  -0.2218,   1.0031)  |q|= 31.466 nm-1  dq:(  0.0058,   0.0446,   0.0444)  Qtot:(  0.9975,   1.7782,   3.0031)
a37: ( -1.0082,  -0.1869,   1.0534)  |q|= 32.225 nm-1  dq:(  0.0062,   0.0446,   0.0443)  Qtot:(  0.9918,   1.8131,   3.0534)
a38: ( -1.0146,  -0.1520,   1.1037)  |q|= 33.032 nm-1  dq:(  0.0066,   0.0446,   0.0443)  Qtot:(  0.9854,   1.8480,   3.1037)
a39: ( -1.0217,  -0.1171,   1.1539)  |q|= 33.882 nm-1  dq:(  0.0070,   0.0446,   0.0442)  Qtot:(  0.9783,   1.8829,   3.1539)
a40: ( -1.0296,  -0.0822,   1.2039)  |q|= 34.772 nm-1  dq:(  0.0074,   0.0446,   0.0441)  Qtot:(  0.9704,   1.9178,   3.2039)
a41: ( -1.0382,  -0.0473,   1.2537)  |q|= 35.698 nm-1  dq:(  0.0079,   0.0446,   0.0440)  Qtot:(  0.9618,   1.9527,   3.2537)

Av.  dq  H(80.0): (-0.0049,0.0254,0.0366)  V(80.0): (-0.0035,-0.0367,0.0251)

HKL values to:  gpi.hkl_pos

Command(BL43LXU):      mv tth 43.6921 th 21.8461 chi -7.7961 phi -19.4072 agaph=80.0 agapv=80.0
```

htth_q()

- Calculate |q| for each analyzer, based on current TTH, MU, GAM

For bl43xu:



```
>>> htth_q()

Qs from 1600 pt mesh. Beam setup 1 = STDM3
SA = 43.6924 deg ->  Qnom = 82.0205 nm-1 at 21.7472 keV  tth = 43.6921
Slit  H: 40.0/4.46/0.256  V: 40.0/4.53/0.259    mm/mrad/deg
mu = -0.1719  gam = 0.0000
Incident Divergence: Div.:  0.60 mrad (0.034 deg  -0.12 nm) H    0.20 mrad V
Qres = sqrt((3.5*rms)^2 + beam_div_in^2)

             A          Q_Av       SLIT_FW          Qres
           a03        78.245         0.450         0.482          nm-1
           a04        79.504         0.449         0.481          nm-1
           a05        80.761         0.448         0.480          nm-1
           a06        82.015         0.447         0.479          nm-1
           a07        83.266         0.446         0.478          nm-1
           a08        84.513         0.444         0.476          nm-1
           a09        85.757         0.443         0.475          nm-1

           a14        78.248         0.443         0.475          nm-1
           a15        79.507         0.443         0.475          nm-1
           a16        80.764         0.442         0.474          nm-1
           a17        82.017         0.441         0.473          nm-1
           a18        83.268         0.440         0.472          nm-1
           a19        84.516         0.439         0.471          nm-1
           a20        85.759         0.438         0.470          nm-1

           a25        78.269         0.437         0.469          nm-1
           a26        79.527         0.436         0.468          nm-1
           a27        80.783         0.436         0.468          nm-1
           a28        82.036         0.435         0.467          nm-1
           a29        83.286         0.434         0.466          nm-1
           a30        84.533         0.433         0.465          nm-1
           a31        85.776         0.432         0.464          nm-1

           a35        78.306         0.430         0.462          nm-1
           a36        79.564         0.430         0.462          nm-1
           a37        80.819         0.430         0.462          nm-1
           a38        82.071         0.429         0.461          nm-1
           a39        83.320         0.428         0.460          nm-1
           a40        84.566         0.427         0.459          nm-1
           a41        85.808         0.426         0.458          nm-1

    Q values written to: gpi.qpos
```