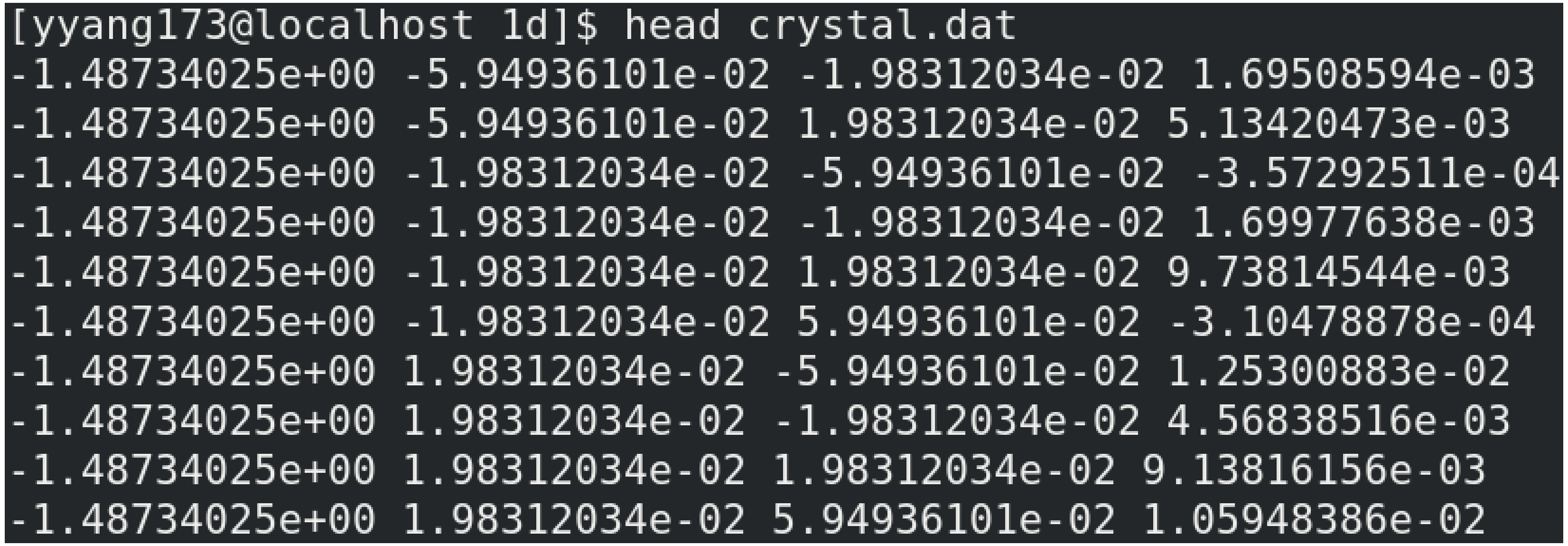
Visualize 3D Scatter Data

Yubo “Paul” Yang, THW, 2019/02/06

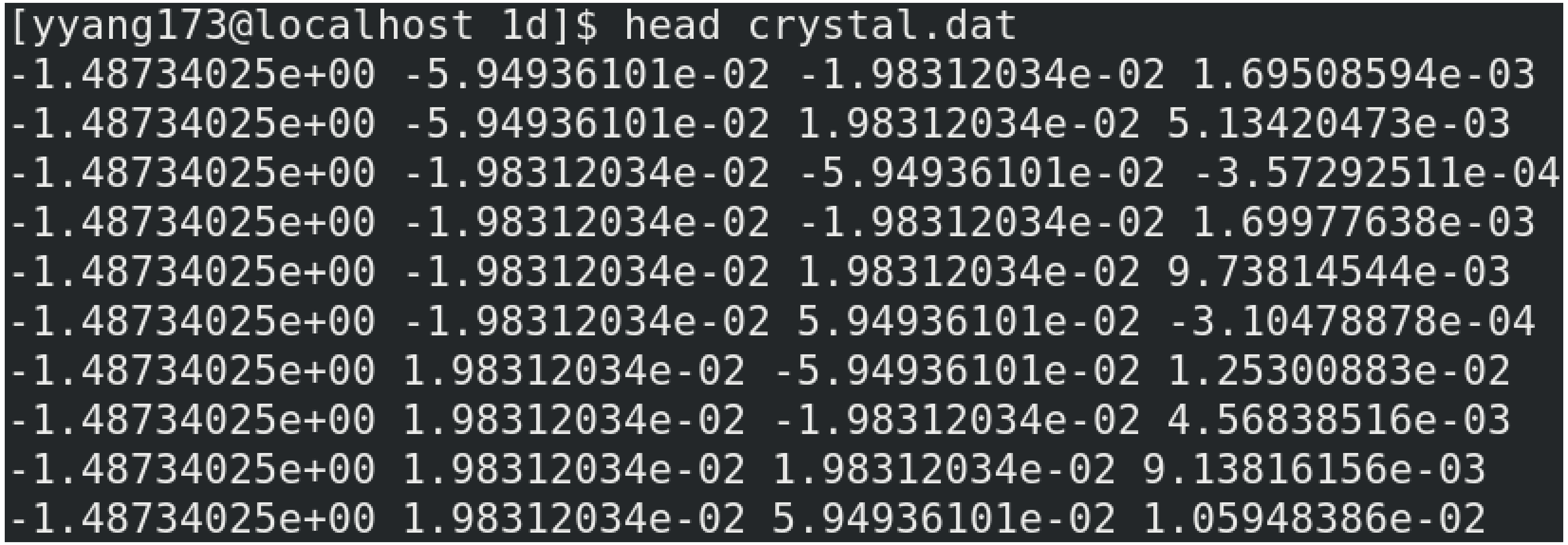


```
[yyang173@localhost 1d]$ head crystal.dat
-1.48734025e+00 -5.94936101e-02 -1.98312034e-02 1.69508594e-03
-1.48734025e+00 -5.94936101e-02 1.98312034e-02 5.13420473e-03
-1.48734025e+00 -1.98312034e-02 -5.94936101e-02 -3.57292511e-04
-1.48734025e+00 -1.98312034e-02 -1.98312034e-02 1.69977638e-03
-1.48734025e+00 -1.98312034e-02 1.98312034e-02 9.73814544e-03
-1.48734025e+00 -1.98312034e-02 5.94936101e-02 -3.10478878e-04
-1.48734025e+00 1.98312034e-02 -5.94936101e-02 1.25300883e-02
-1.48734025e+00 1.98312034e-02 -1.98312034e-02 4.56838516e-03
-1.48734025e+00 1.98312034e-02 1.98312034e-02 9.13816156e-03
-1.48734025e+00 1.98312034e-02 5.94936101e-02 1.05948386e-02
```

What is 3D scatter data?

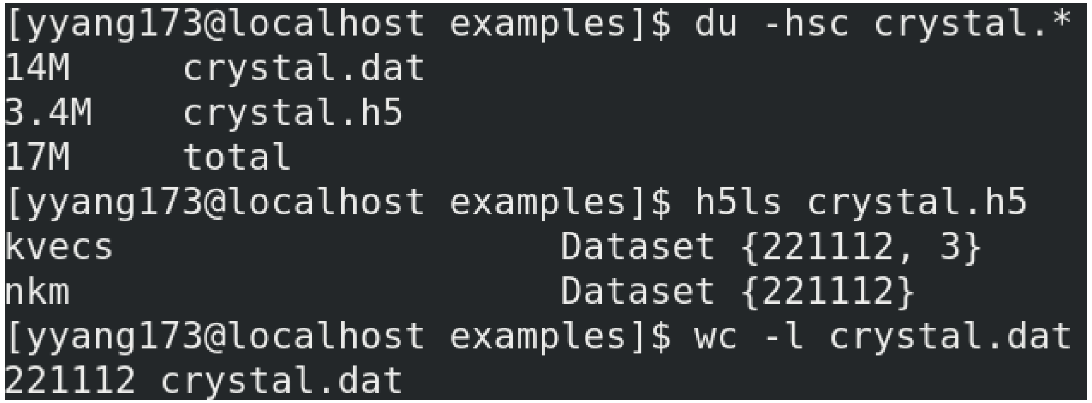
3D scatter data is a finite collection of 3-dimensional coordinates, each associated with a value.

x **y** **z** **value**



```
[yyang173@localhost 1d]$ head crystal.dat
-1.48734025e+00 -5.94936101e-02 -1.98312034e-02 1.69508594e-03
-1.48734025e+00 -5.94936101e-02 1.98312034e-02 5.13420473e-03
-1.48734025e+00 -1.98312034e-02 -5.94936101e-02 -3.57292511e-04
-1.48734025e+00 -1.98312034e-02 -1.98312034e-02 1.69977638e-03
-1.48734025e+00 -1.98312034e-02 1.98312034e-02 9.73814544e-03
-1.48734025e+00 -1.98312034e-02 5.94936101e-02 -3.10478878e-04
-1.48734025e+00 1.98312034e-02 -5.94936101e-02 1.25300883e-02
-1.48734025e+00 1.98312034e-02 -1.98312034e-02 4.56838516e-03
-1.48734025e+00 1.98312034e-02 1.98312034e-02 9.13816156e-03
-1.48734025e+00 1.98312034e-02 5.94936101e-02 1.05948386e-02
```

Pro. Tip:
Store 3D scattered data in hdf5 file with a **compression filter**.
Not in an ASCII file as shown above.



```
[yyang173@localhost examples]$ du -hsc crystal.*
14M     crystal.dat
3.4M    crystal.h5
17M     total
[yyang173@localhost examples]$ h5ls crystal.h5
kvecs                    Dataset {221112, 3}
nkm                      Dataset {221112}
[yyang173@localhost examples]$ wc -l crystal.dat
221112 crystal.dat
```

Why visualize 3D scatter data?

3D scatter data is **prevalent** and easily processed by 2019 hardware*

General examples:

- Heat distribution in a room (discrete measurements)
- Mineral concentration in the mantle (discrete measurements)
- Audience input 1
- Audience input 2

Electronic structure examples:

- Momentum distribution of the electrons $n(\boldsymbol{k})$
- Static structure factor $S(\boldsymbol{k})$
- Band dispersion $\epsilon(\boldsymbol{k})$
- Atomic/molecular orbital $\phi(\boldsymbol{r})$
- Electronic density $\rho(\boldsymbol{r})$

*the same cannot be said about software (e.g. matplotlib)

How to visualize 3D scatter data?

Example: QMC momentum distribution of valence electrons in solid and liquid lithium

1D View:

- Spherical average
- 1D slices

2D View:

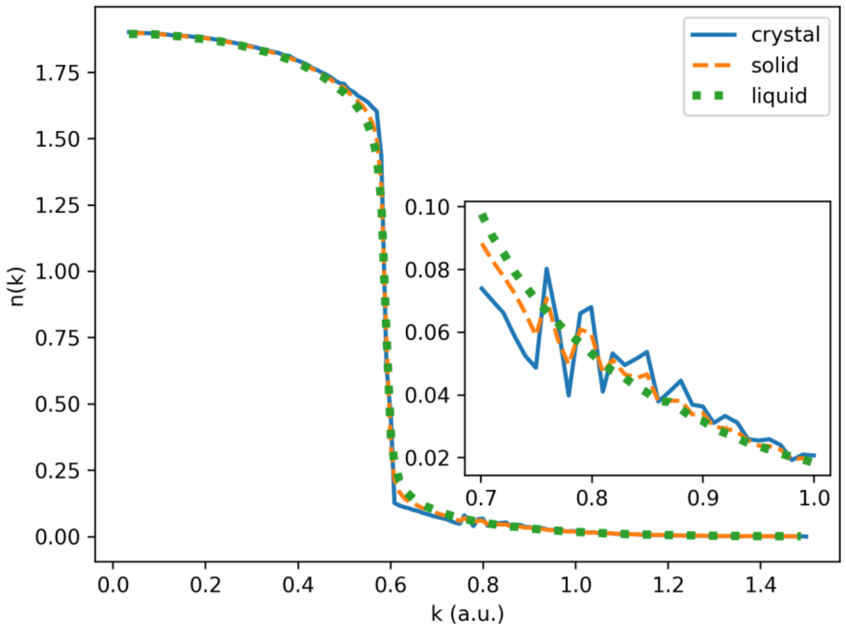
- 2D slices
- Slice and scan (animation)

3D View:

- Isosurface
- Isosurface and scan (animation)
- Colored scatter
- Scatter and rotate (animation)

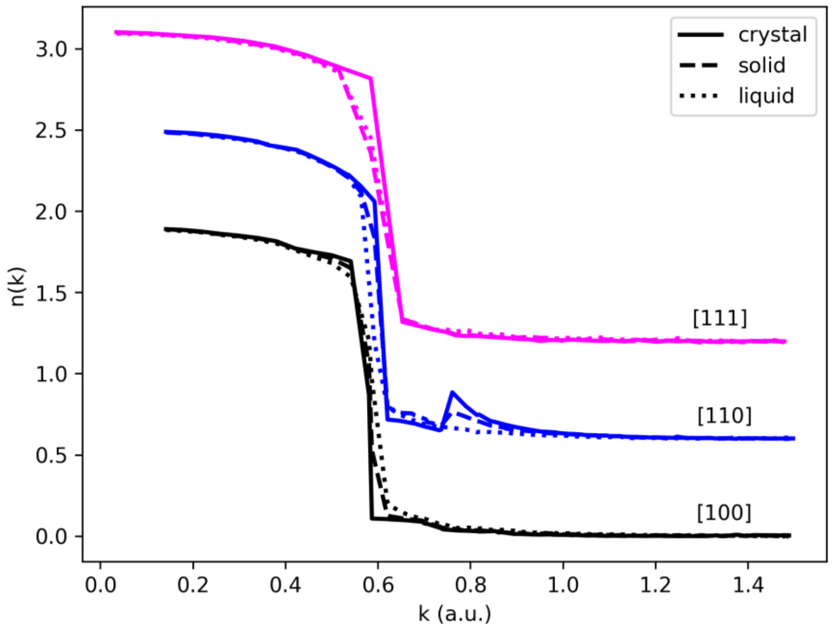
1D View

Spherical Average



- Liquid n(k) is smooth
- Solid n(k) has extra wiggles at $k \approx 0.8$
- Crystal n(k) has larger wiggles at the same place as the solid and is sharper at $k \approx 0.6$

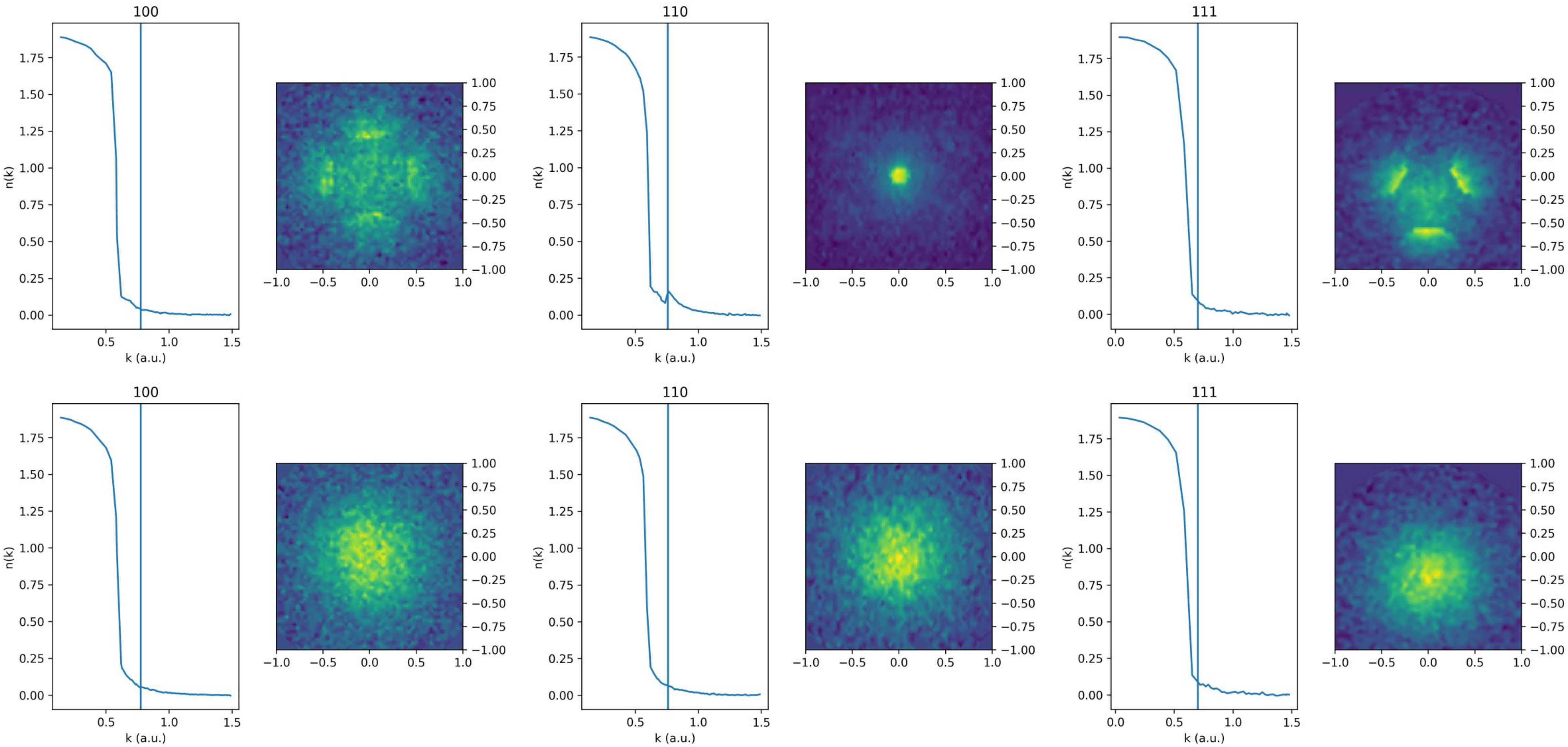
1D Slices



- Liquid n(k) is smooth in all 3 directions
- Solid n(k) has a bump along [110] only
- Crystal n(k) is similar to solid in all 3 directions, but sharper

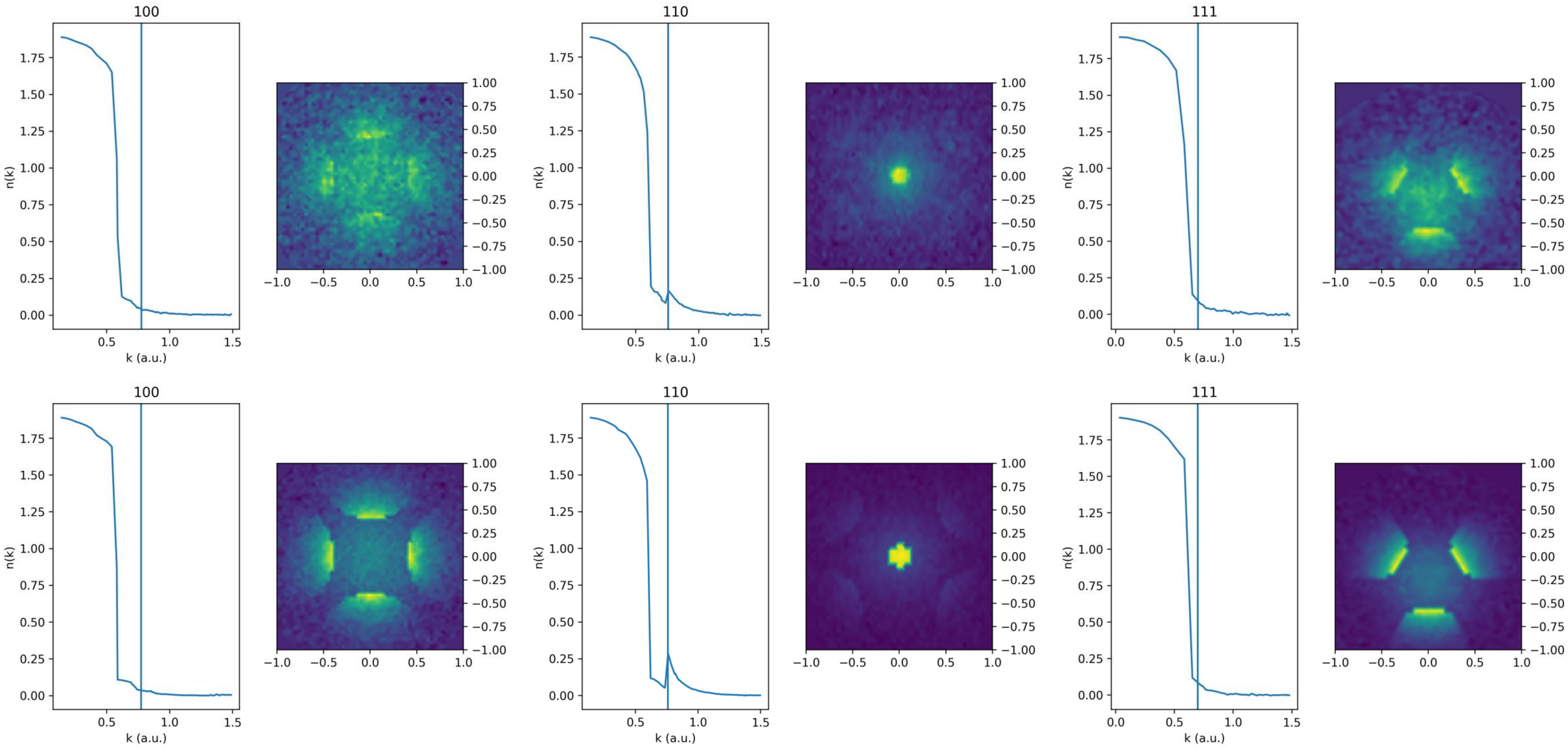
plt.plot

2D View slices: solid v.s. liquid



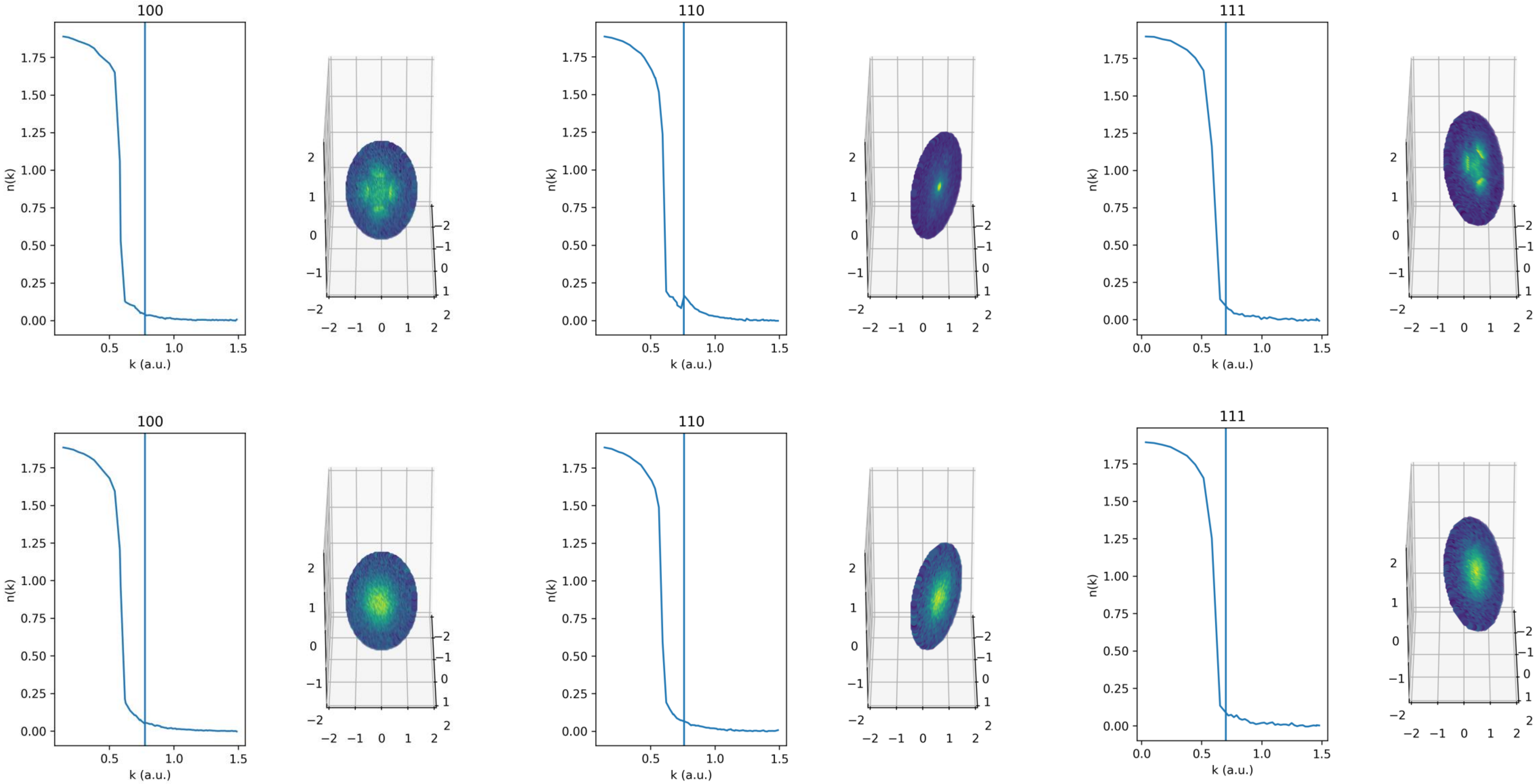
plt.pcolormesh

2D View slices: solid v.s. crystal



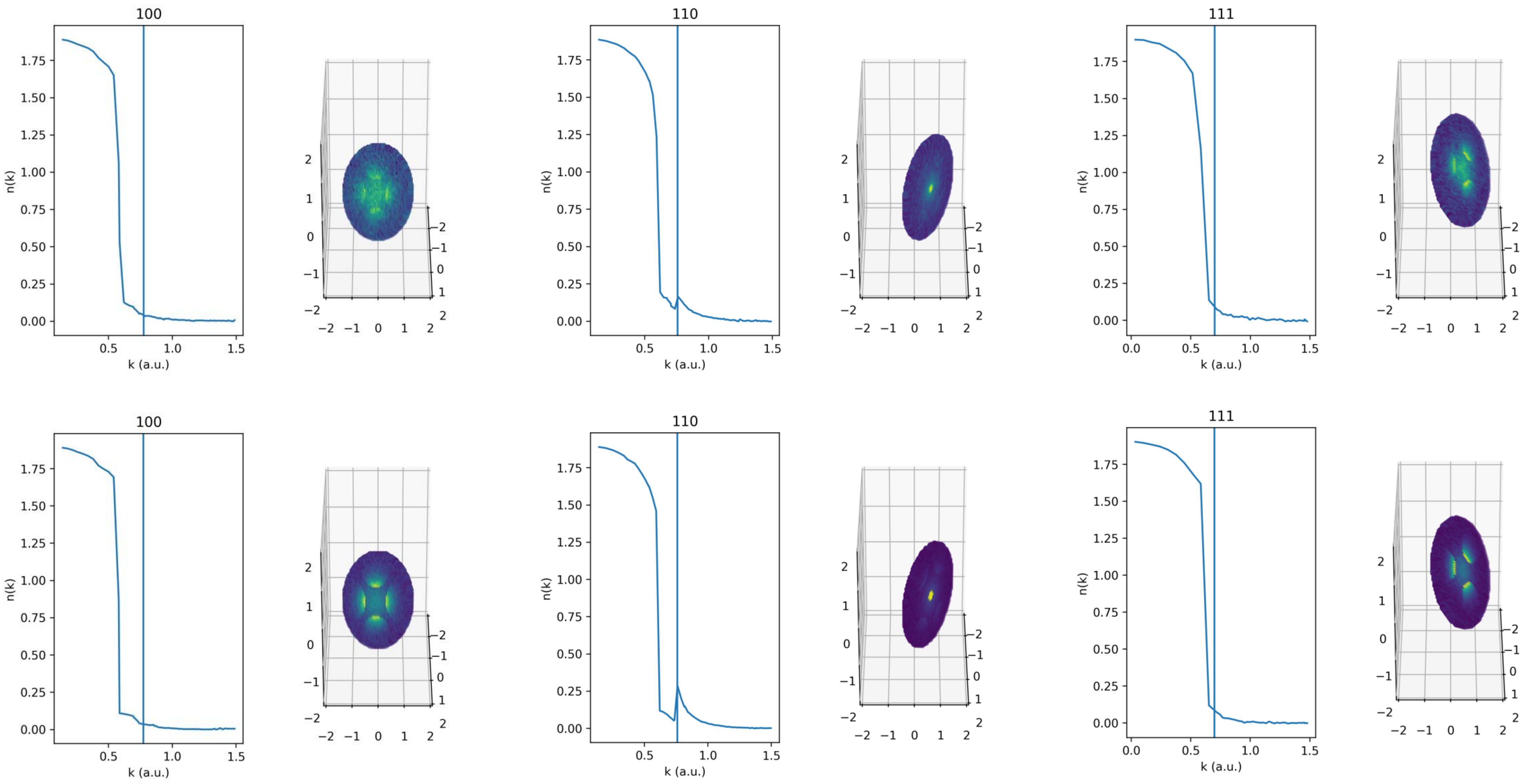
plt.pcolormesh

2D* View slices: solid v.s. liquid



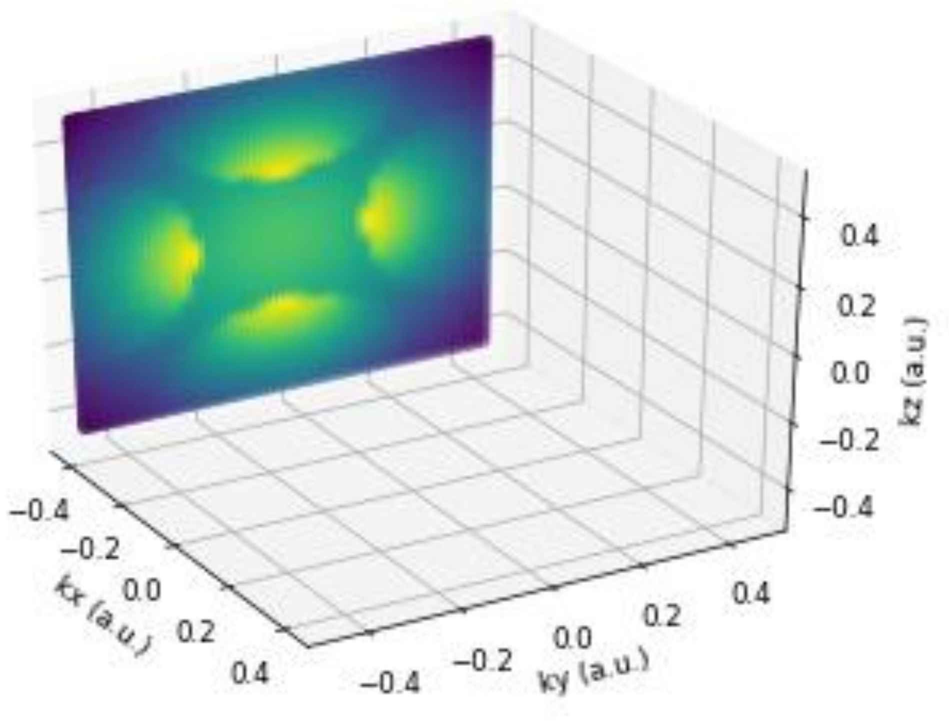
*Axes3D.scatter

2D* View slices: solid v.s. crystal



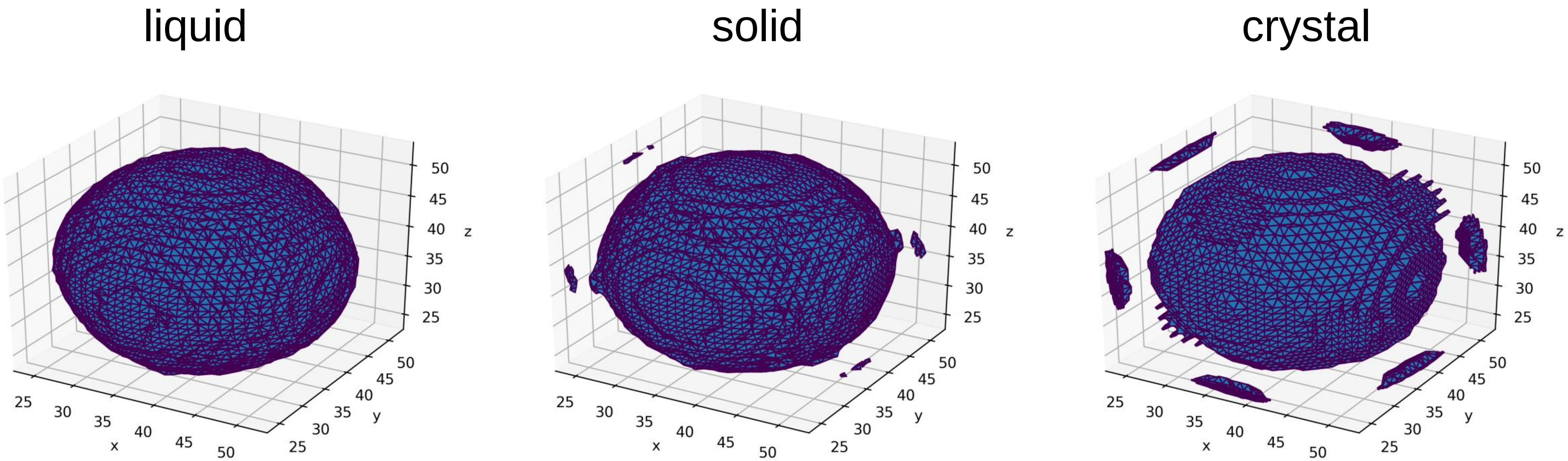
*Axes3D.scatter

2D View slice and scan: crystal



Axes3D.scatter and plt.animation.ArtistAnimation

3D View: isosurface



liquid

solid

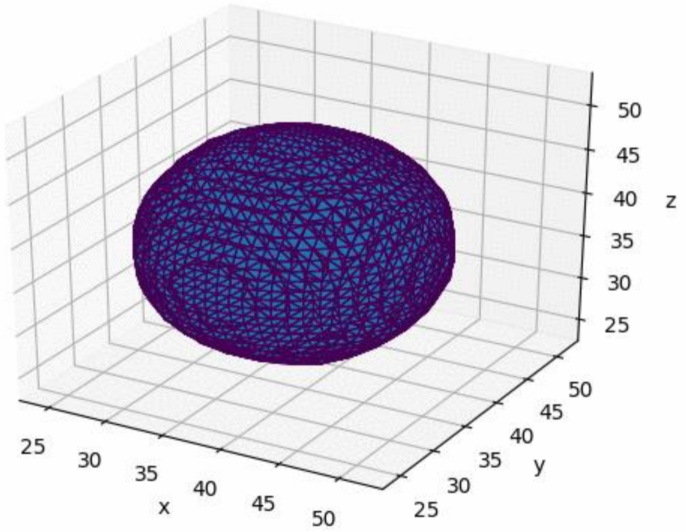
crystal

See algorithm interest group (AIG) post for details.

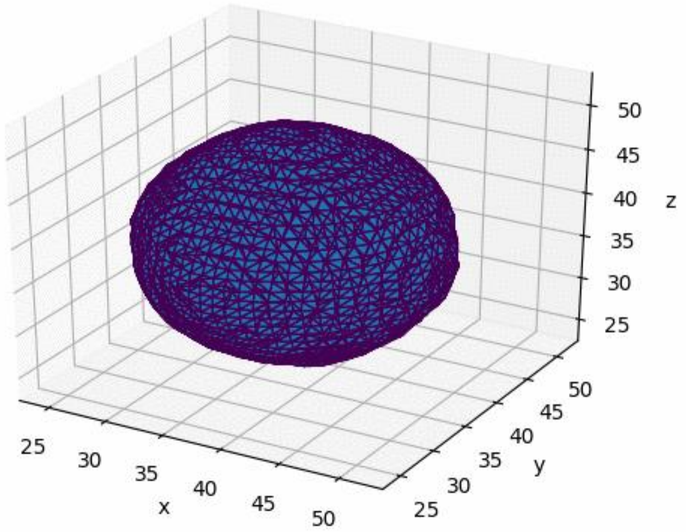
mpl_toolkits.mplot3d.art3d.Poly3DCollection and skimage.measure.marching_cubes_lewiner

3D View: isosurface and scan

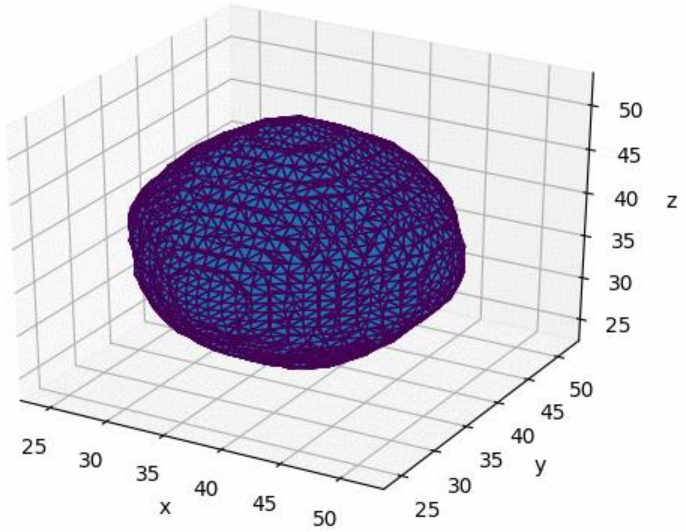
liquid



solid

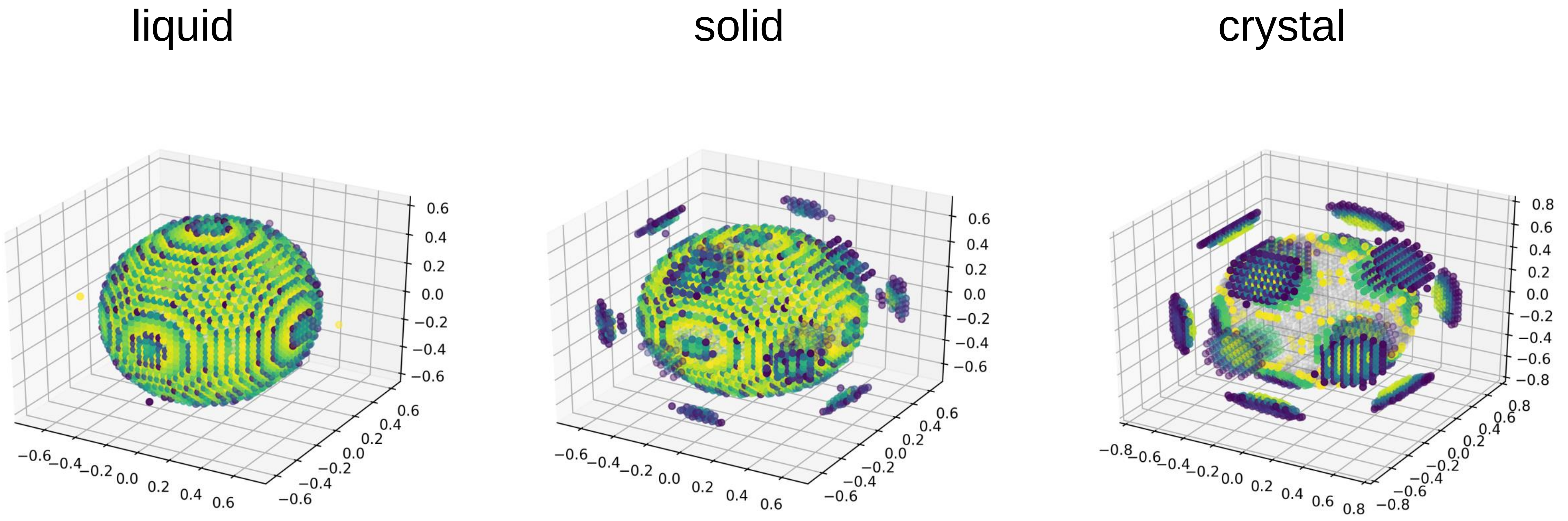


crystal



mpl_toolkits.mplot3d.art3d.Poly3DCollection
skimage.measure.marching_cubes_lewiner
matplotlib.animation.ArtistAnimation

3D View: colored scatter



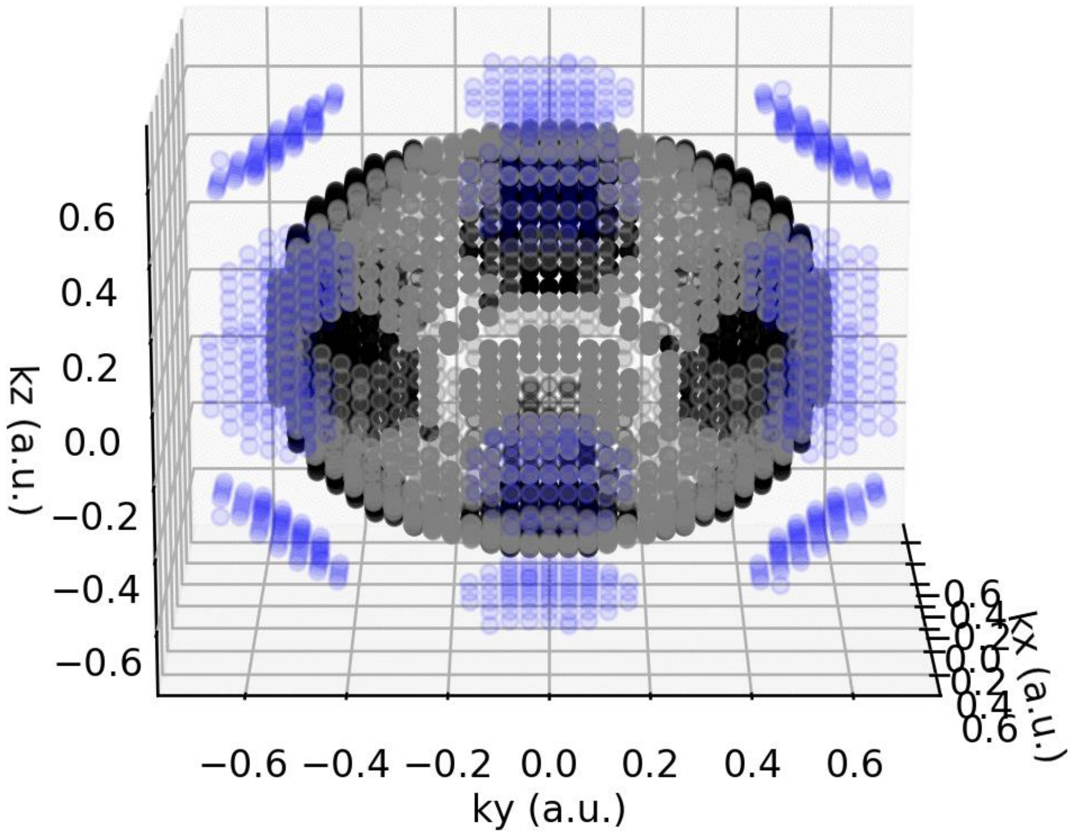
liquid

solid

crystal

Axes3D.scatter

3D View: scatter and rotate



A Great Programmer Steals

Export to Gaussian cube file: readable by

- VESTA
- XCrySDen
- VMD

Export to wavefront obj file: readable by

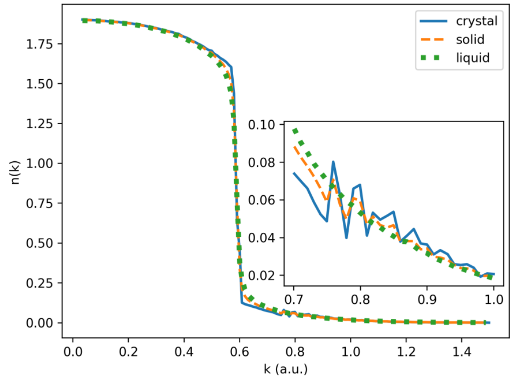
- Blender
- Unreal Engine*
- Unity

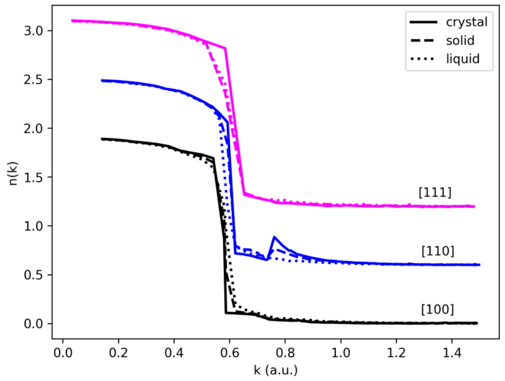
*see example at the hacker within (THW) post

Conclusions: 3D Scatter

1D View:

- Spherical average
- 1D slices



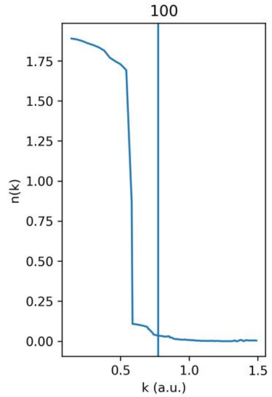


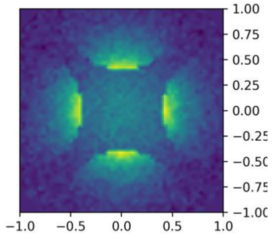
most quantitative

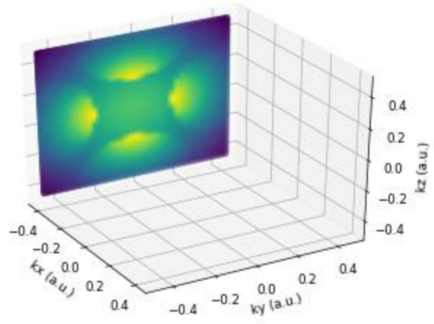
most illustrative

2D View:

- 2D slices
- Slice and scan (animation)

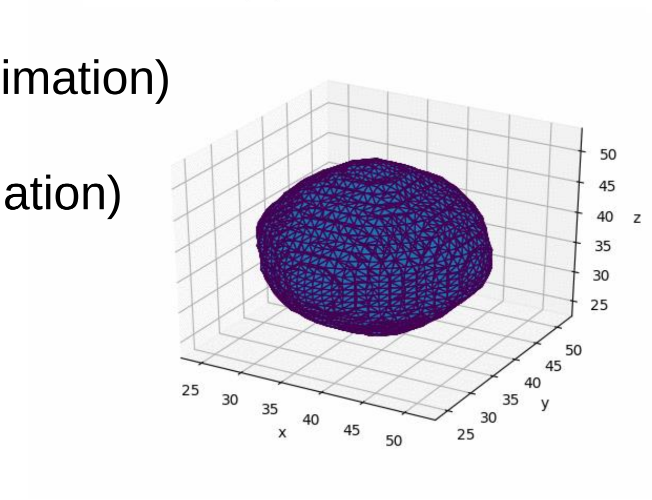


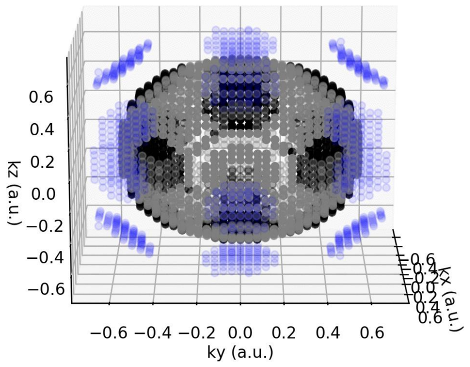




3D View:

- Isosurface
- Isosurface and scan (animation)
- Colored scatter
- Scatter and rotate (animation)





3D Codes:

I have written some simple wrappers for using matplotlib 3D library to visualize 3D scatter data.

See:

- figax3d()
- color_scatter
- Isosurf
- write_gaussian_cube

in qharv.inspect.volumetric

Usage example: isosurface using marching cubes