**Python Data Shell – GSECARS CTR data integration code**

**Tom Trainor, Matt Newville, Craig Biwer, Peter Eng, Frank Heberling, Joanne Stubbs**

**Major capabilities and modules that enable them:**

**Gathers images and data from SPEC files into HDF5 files**

spectohdf.py (and variants)

auto\_master.py (and variants)

mastertoproject.py

hdf\_data.py

image\_data.py

**Provides GUIs for selecting SPEC scans, viewing images, and interactively integrating data**

wxFilter.py

run\_filter.py

filtertools.py

wxIntegrator.py

run\_integrator.py

wxHDFtoTree.py

image\_data.py

file\_locker.py

**Background subtraction on image-by-image basis**

background.py

**Io normalization**

ctr\_data.py

**Lorentz and polarization corrections**

ctr\_data.py

gonio\_psic.py

lattice.py

**Beam foot print and spill-off corrections**

ctr\_data.py

active\_area.py

polygon.py

mathutil.py

gonio\_psic.py

lattice.py

**Correction for attenuators in beam path**

ctr\_data.py

**Calculates error bars based on Poisson statistics**

ctr\_data.py

**Outputs ASCII files containing HKL, structure factors, error bars, “raw” intensities, energies for resonant scans, etc.**

wxIntegrator.py

**All code is in python and so in theory platform independent; can also provide Windows executables**

setup\_py2exe.py

inno\_setup\_distonly2.iss

**Dependencies and versions used:**

Python 2.7.2 (32-bit)

h5py 2.0.1

matplotlib 1.0.1

numpy 1.6.1

python imaging library (PIL) 1.1.7

scipy 0.10.0b2

wxPython 2.8.12

py2exe 0.6.9 (only needed for generating Windows executables)

inno setup 5.5.5 (only needed for generating Windows executables)

HDFView 2.9, 32-bit (only needed to view hdf5 outside Python Data Shell)

**Capabilities not yet implemented but desired:**

3D integration and binning of low-L data where rods are nearly tangent to Ewald sphere

Comparing scans – overplotting, math operations (currently done in GenX after ASCII output)

Symmetry equivalent averaging, weighting, and systematic error estimation (currently done in AVE)

Write L-Bragg and delta-L values for Robinson roughness model (currently done manually)

Geometries other than PSIC

**General improvements desired**:

Update python dependencies

Make data structure more readily customizable

Make code less CTR-centric (useful for other types of diffraction data)