BLAND UI: User-Friendly Neutron Diffraction Analysis

Nathan Super

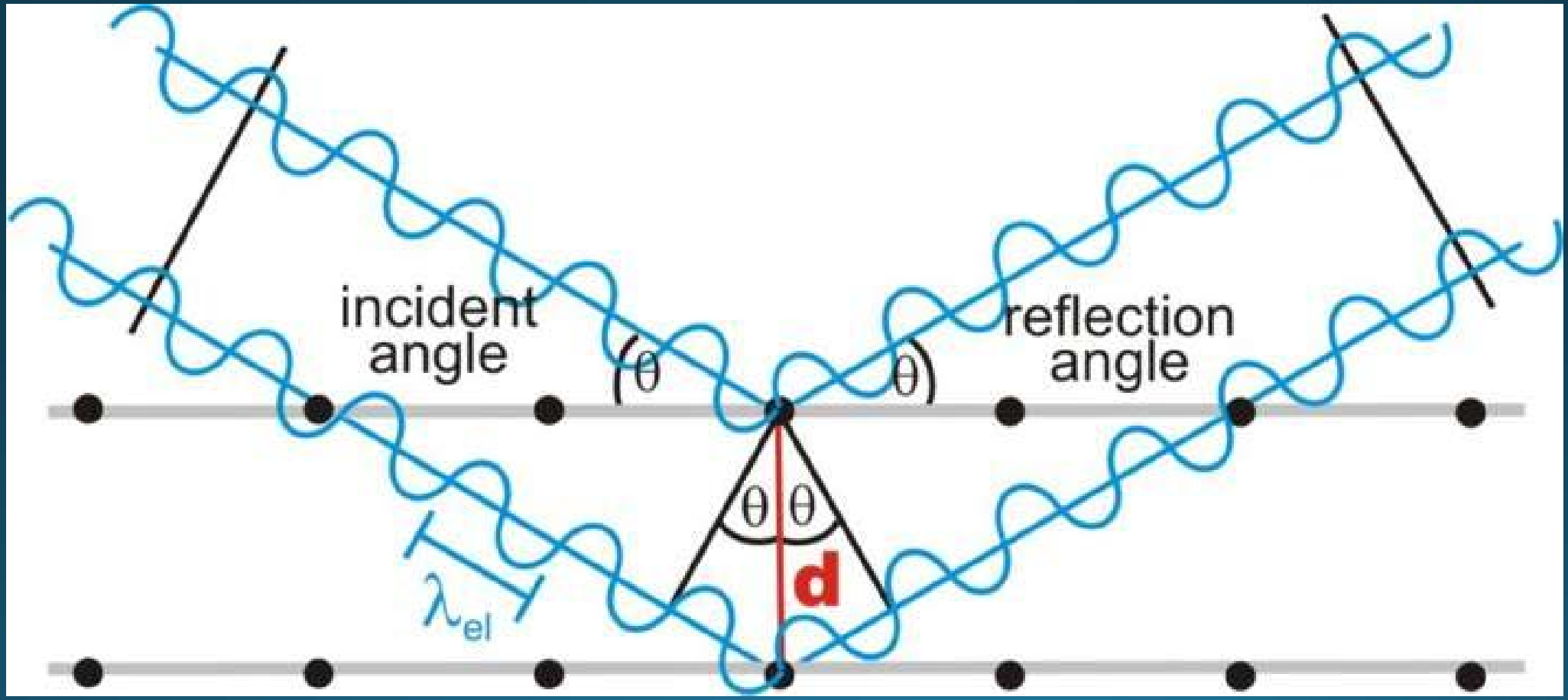
William Ratcliff (NCNR)
Joseph Lesniewski
Dylan Quintana
Steven Disseler (NCNR)
Paul Kienzle (NCNR)

Inverse Fitting Problem

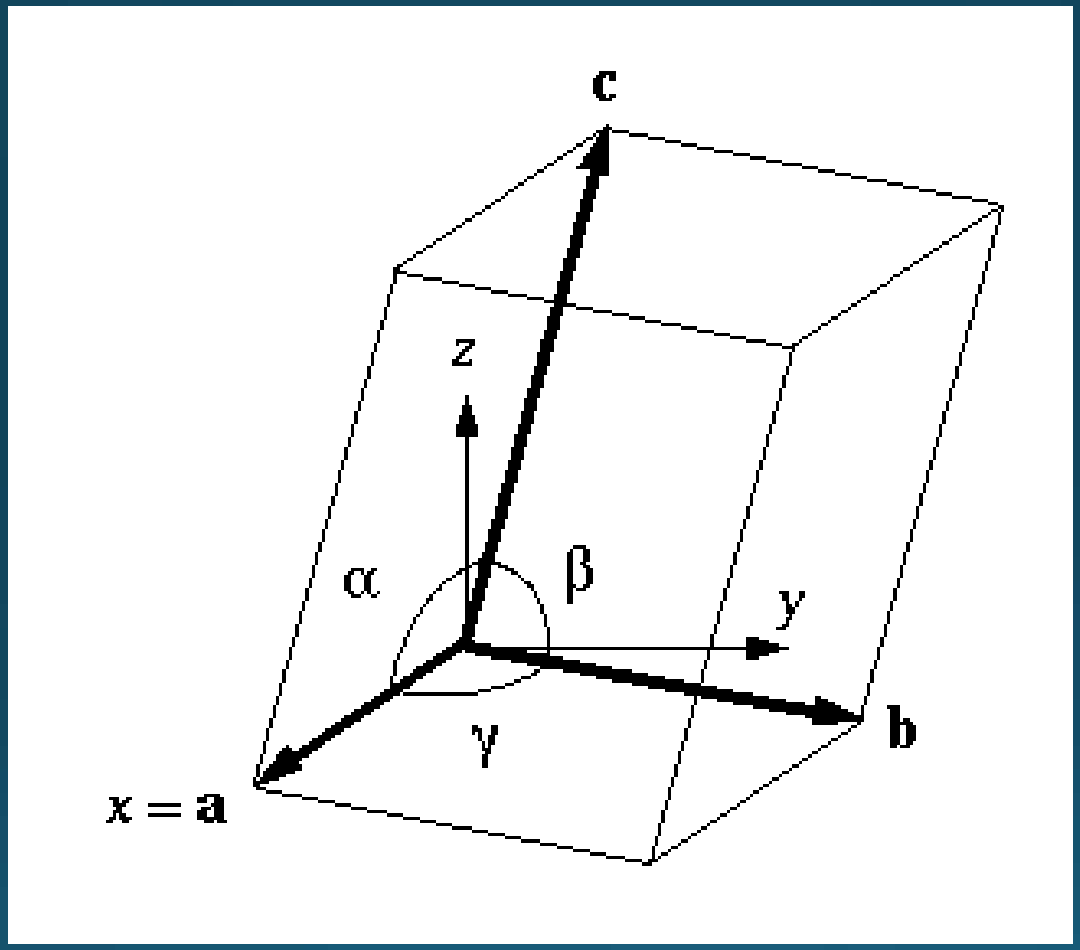
- Goal: Data → Model
- Data is intensity pattern
- Model is crystal system

Bragg Diffraction

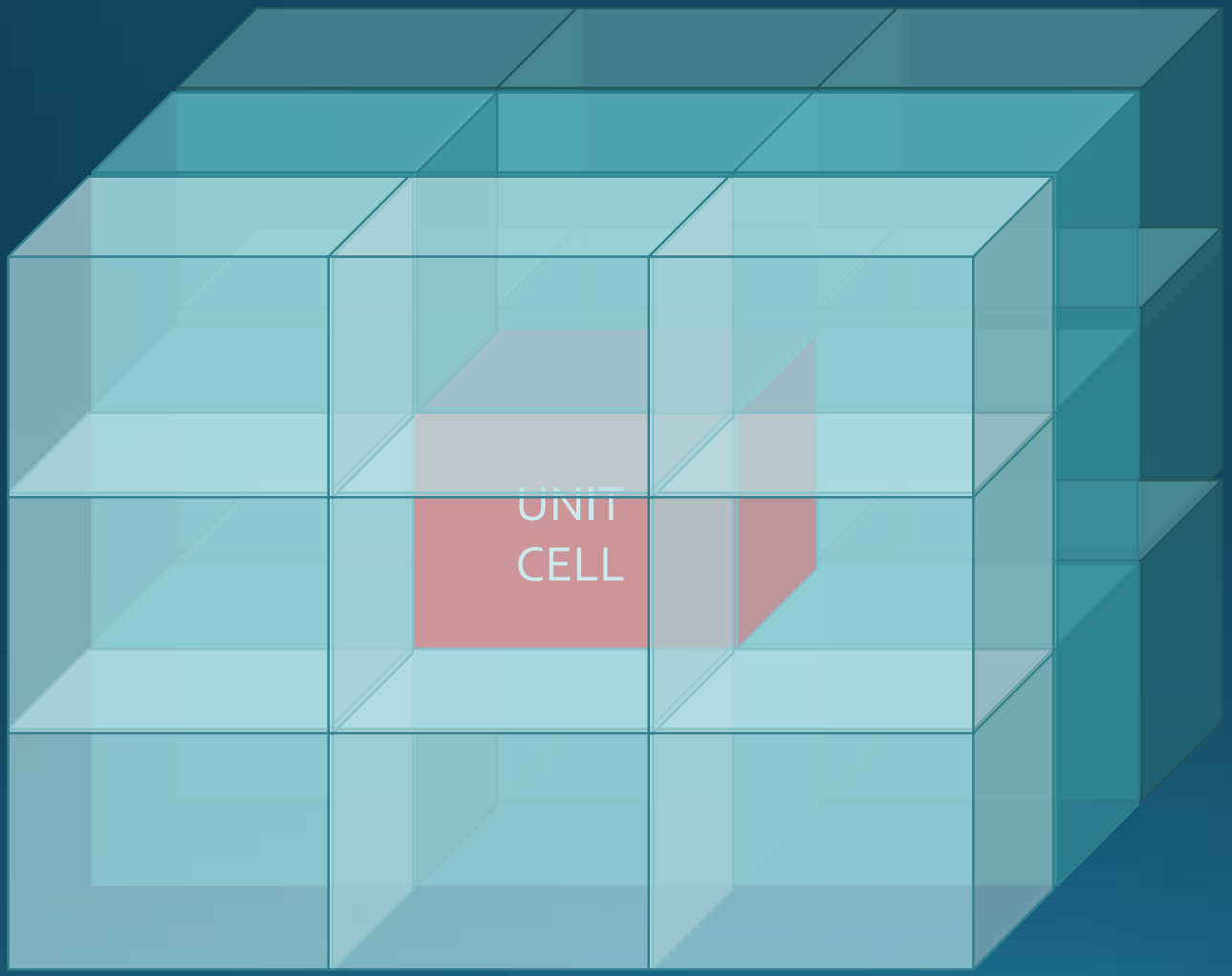
Bragg's Law: $2d \sin\theta = n\lambda$



Crystallography



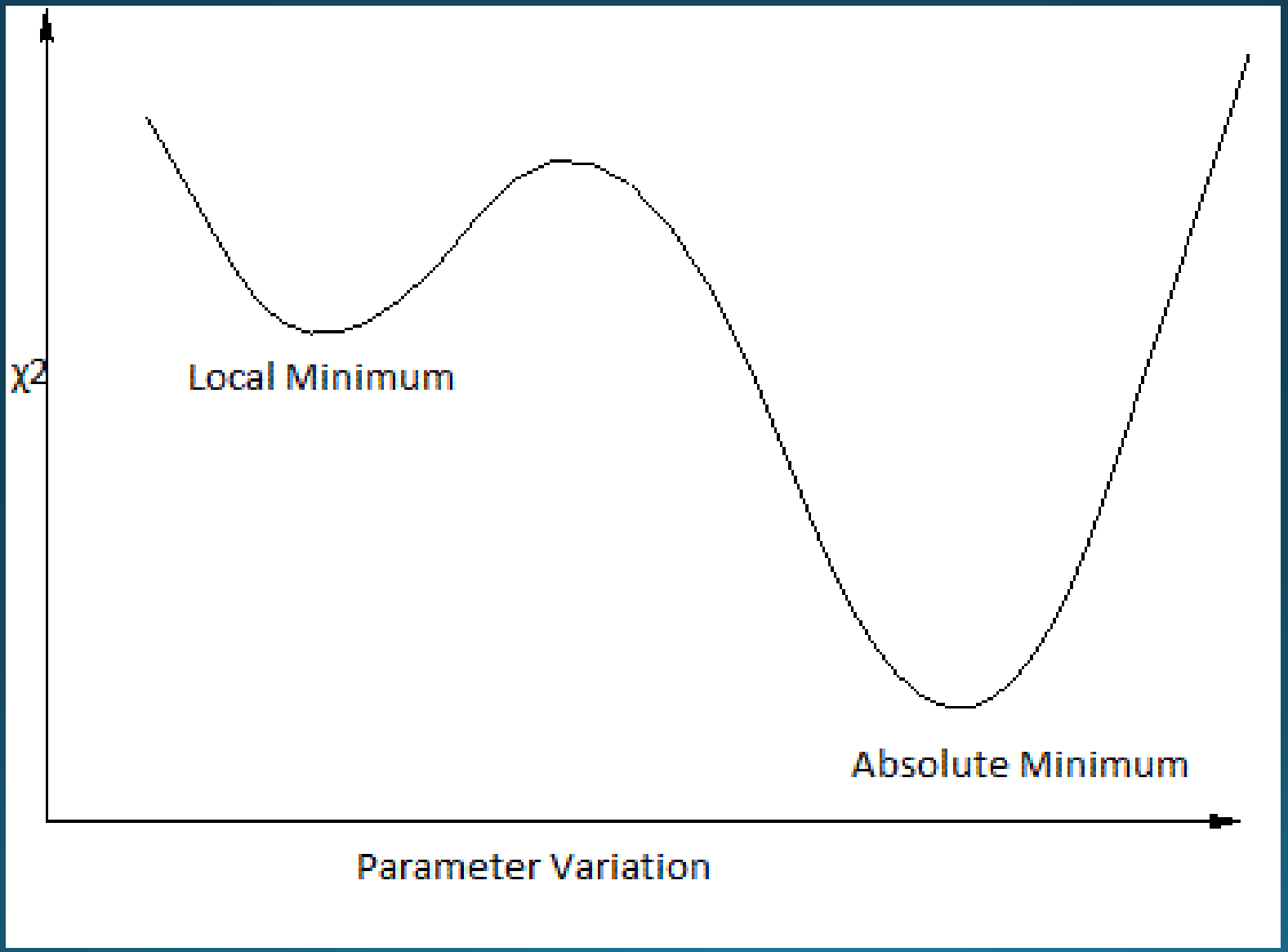
- Unit Cell
- Translational, Rotational Symmetry


UNIT
CELL

Traditional Approach

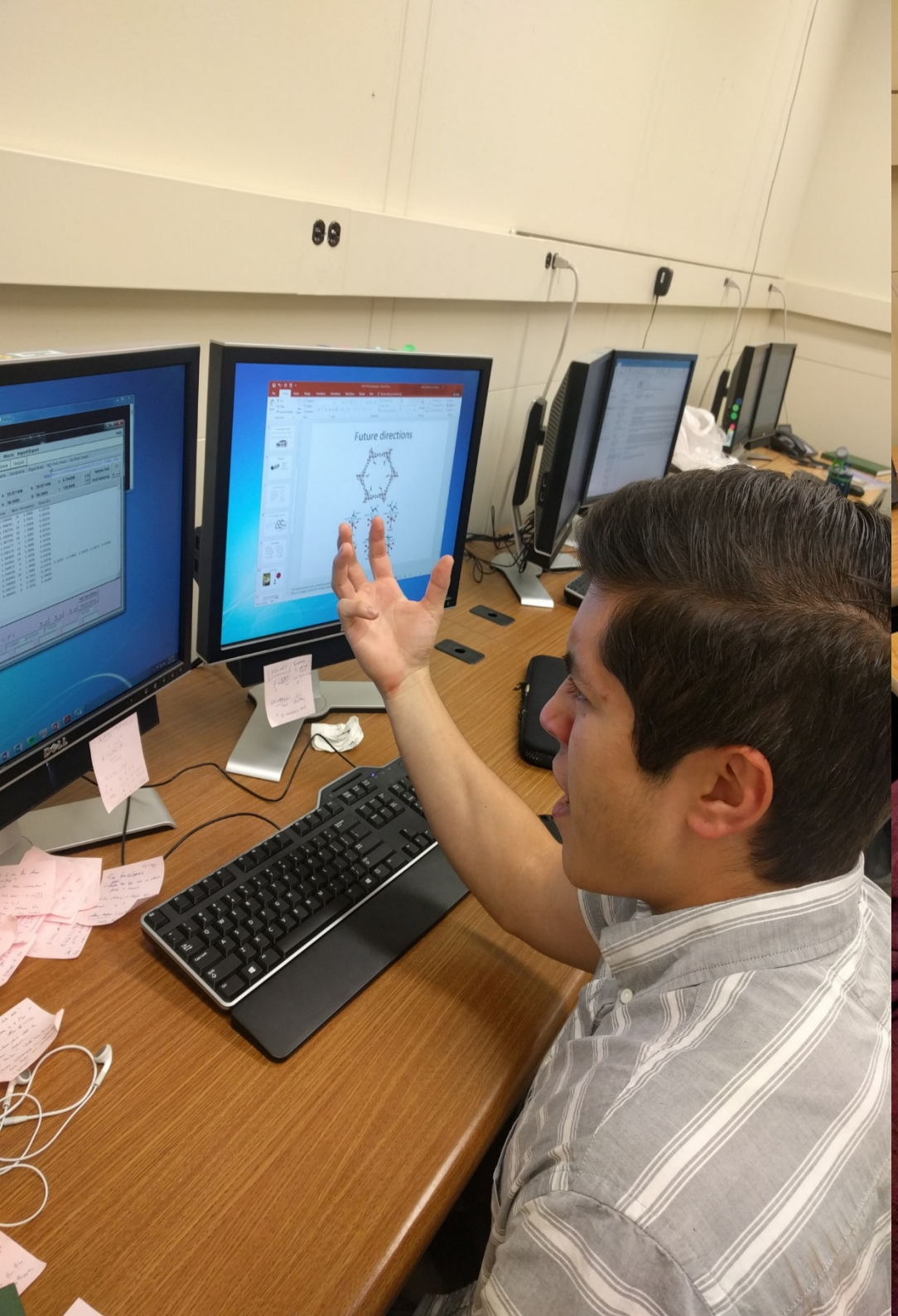
- Reitveld Refinement
- Minimizes χ^2
- Gets stuck in local minima

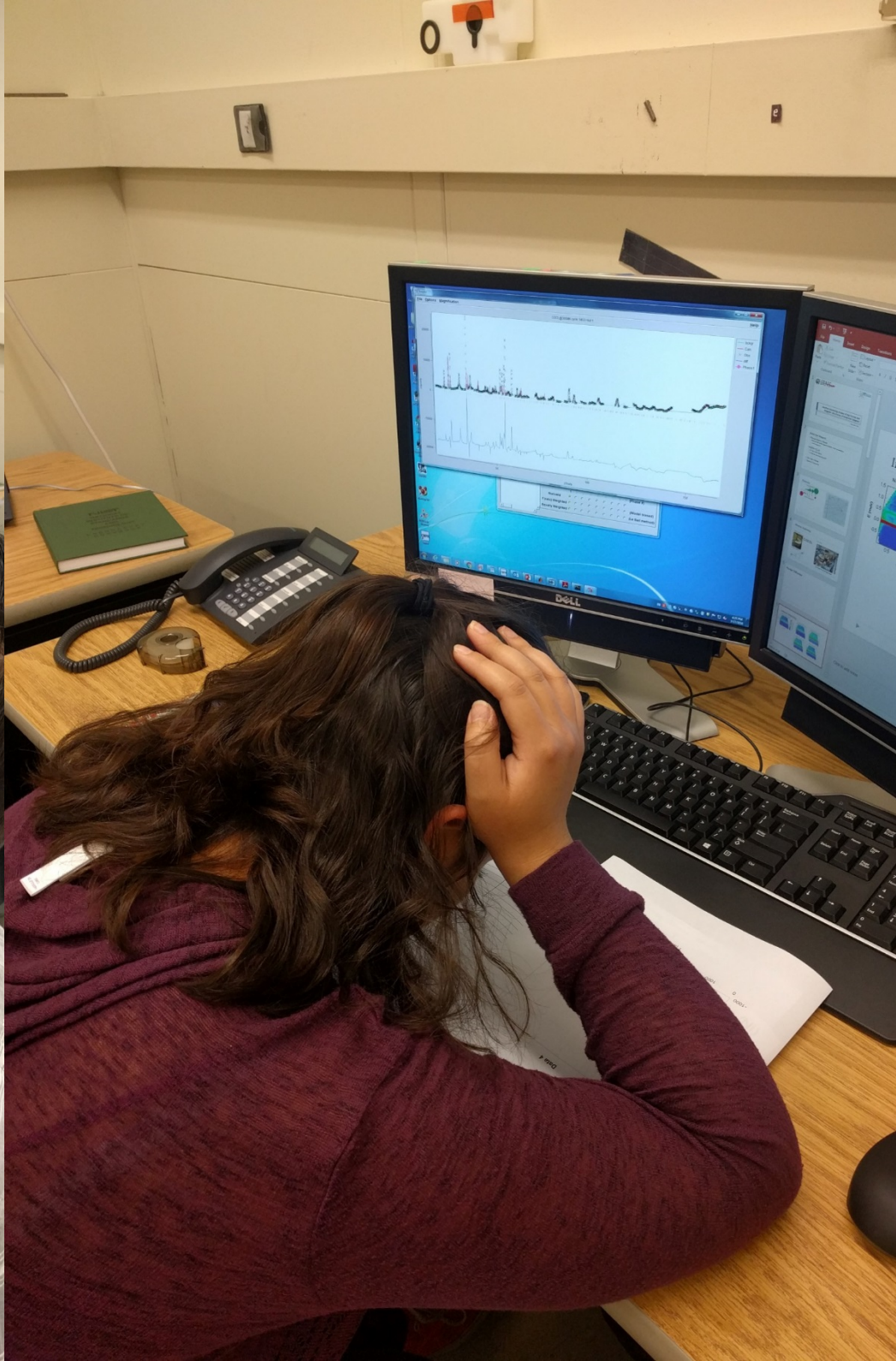
Gauss-Newton Algorithm



Traditional Approach

- Reitveld Refinement
- Minimizes residuals
- Gets stuck in local minima
- Requires good guess of model parameters
- Not guaranteed to converge


Future directions



Bayesian Alternative

- Maximizes probability
- Algorithms used efficiently sample parameter space
- Parallel sampling improves convergence, chance of correct solution

BUMPS

Bayesian Uncertainty Modeling of Parametric Systems

```
step 10206 cost 3.85175379008
                                    O1 x ....|.....   0.909055 in (-0.1,2)
                                    O1 z .....|....   0.595744 in (-0.5,1.6)
                                    O2 x .....|....   0.192831 in (-0.9,1.2)
                                    O2 z ....|.....   0.542582 in (-0.5,1.6)
                                    O3 x .....|....   0.081153 in (-1,1.1)
                                    O3 y ....|.....  0.0282213 in (-1,1.1)
                                    O3 z ....|.....   0.807125 in (-0.2,1.9)
                                    Pb B |.........   0.374014 in (0,10)
                                    Pb x .....|....    0.18723 in (-0.9,1.2)
                                    Pb z .....|....   0.167179 in (-0.9,1.2)
                                     S x .....|....  0.0648723 in (-1,1.1)
                                     S z .....|....   0.685412 in (-0.4,1.7)
                                   base .....|....    213.475 in (-80,430)
                                      a .....|....    8.47816 in (7.9,9)
                                      b .....|....    5.39688 in (4.8,5.9)
                                      c .....|....    6.95831 in (6.4,7.5)
                                    eta ....|.....   0.451881 in (0,1)
                                  scale .|........    1.14508 in (0,10)
                                      u |.........   0.155385 in (0,2)
                                      v ........|.  -0.361573 in (-2,0)
                                      w .|........   0.374667 in (0,2)
                                   zero ...|......  -0.138759 in (-0.3,0.11)
step 10207 cost 3.85175379008
```

BLAND:
Bayesian Library for Analysis of Neutron Diffraction

Combines BUMPS with CrysFML (Fortran crystallography library)

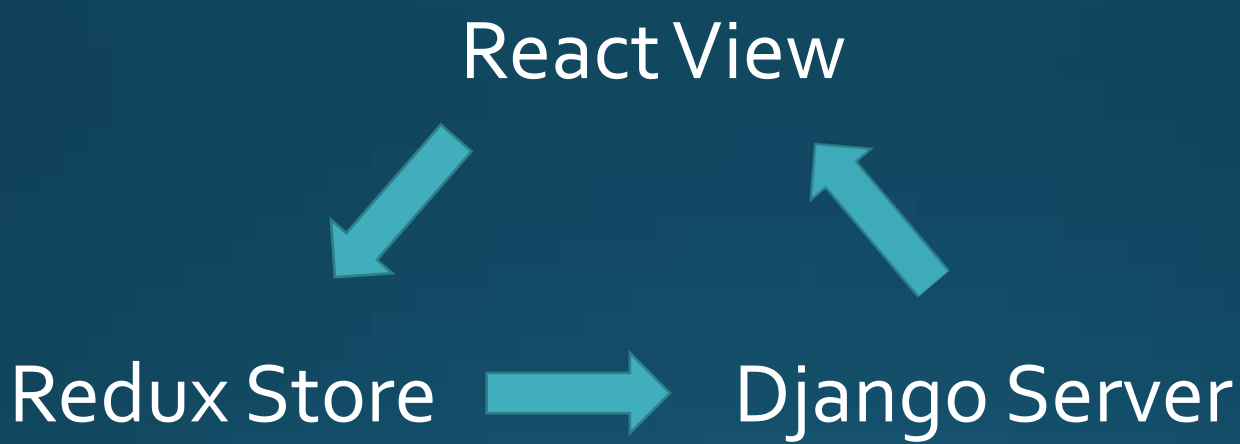
Goals

- ✓ Automated Global Fitting
- ✓ Decrease need for intuition
- ✓ Reliability
- ✓ Adaptability
- ❑ Web interface – My job

User Interface

- ReactJS
- Django
- CrysFML

Architecture



User Interface

- ReactJS
- Django
- CrysFML
- No more need to write own scripts to use BLAND

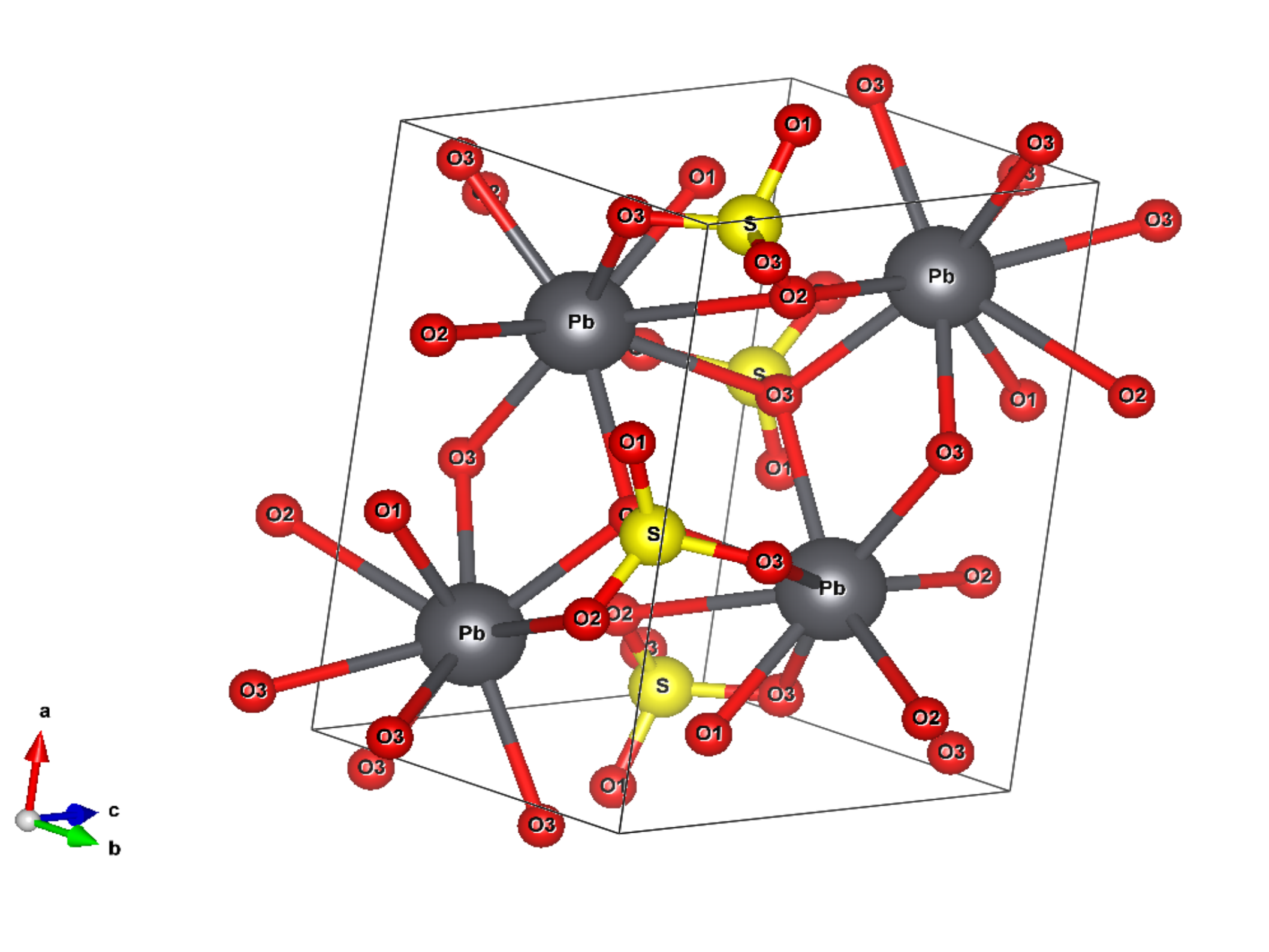
```
np.seterr(divide="ignore", invalid="ignore")

DATAPATH = os.path.dirname(os.path.abspath(__file__))
observedFile = os.path.join(DATAPATH,r"MnWO4_nuclear_5K.int")
infoFile = os.path.join(DATAPATH,r"AF4_5k.cfl")
magfile = os.path.join(DATAPATH, r"MnWO4_magneticAF4_5K.int")

(spaceGroup, crystalCell, magAtomList, symmetry) = H.readMagInfo(infoFile)
spaceGroup, crystalCell, atomList = H.readInfo(infoFile)
exclusions = []
# return wavelength, refList, sfs2, error, two-theta, and four-circle parameters
wavelength, refList, sfs2, error = S.readIntFile(observedFile, kind="int", cell=crystalCell)
magrefList, magsfs2, magerror = S.readMagIntFile(magfile, cell=crystalCell)
tt = [H.twoTheta(H.calcS(crystalCell, ref.hkl), wavelength) for ref in refList]
sfs2 = list(sfs2)
sfs2.extend(magsfs2)
sfs2 = np.array(sfs2)
tt.extend([H.twoTheta(H.calcS(crystalCell, ref.hkl), wavelength) for ref in magrefList])
error = list(error)
error.extend(list(magerror))
error2 = [item for item in error]
backg = None
basisSymmetry = copy(symmetry)

def fit():
    cell = Mod.makeCell(crystalCell, spaceGroup.xtalSystem)
    #cell.a.pm(5.0)
    #cell.c.pm(5.0)
    m = S.Model(tt, sfs2, backg, wavelength, spaceGroup, cell,
                (atomList, magAtomList), exclusions, magnetic=True,
                symmetry=symmetry, newSymmetry=basisSymmetry, scale=1.00,
                error=error, hkls=refList, extinction=[0,0,0,0])
    m.scale.range(0,2000)
    #m.base.pm(1000)
    for ext in m.extinctions:
        ext.range(0,100.0)
    for atomModel in m.atomListModel.atomModels:
        #atomModel.x.range(0,1)
        #atomModel.y.range(0,1)
        #atomModel.z.range(0,1)
        #atomModel.B.range(0,10)
        if atomModel.magnetic:
            for coeff in atomModel.coeffs:
                coeff.range(-20, 20)
                #coeff.range(0,5)
    M = bumps.FitProblem(m)
    M.model_update()
    return M

def main():
    cell = crystalCell
    S.diffPatternXtal(infoFile=infoFile, cell=cell, scale=186.0, tt=tt,
                      obsIntensity=sfs2, wavelength=wavelength,
                      plot=True, residuals=True, error=error2,
                      info=True, base=0, refList=refList, extinctions=[ 3.607,2.156,4.335,0.4110,0.2721,0.0], magAtomList=magAtomList, magnetic=True)
if __name__ == "__main__":
    # program run normally
    main()
else:
    # called using bumps
    import bumps.names as bumps
    problem = fit()
```


a
c
b
Pb
S
O1
O2
O3

BLAND UI

Nathan

localhost:8001/bland/

Welcome to BLAND UI

Help | Instrument | Models | Fit | Results

- Create a crystal model and calculate intensity and structure factor spectra.
- Upload observed intensity data and get a graph of the residuals.
- Fit model to match observed data.
- Under instrument, choose mode, then upload data, fill in at least wavelength, u, v, w.
- Under model, upload a .cif file to fill in crystal model, or enter in parameters by hand.
- Press calculate to get structure factors.
- Under results, press button to show graphs.
- Under fit, give range to fit each parameter, then press fit to run.
- This will open a tab to monitor the status of the fit.
- Access this status page at any time, using the url of the page (http://localhost:8001/status/{token}).
- Save the updated parameters from that page, then upload the file below.
- When the fit finishes, save the plots generated.

Upload a parameter file from a previous fit.

BLAND UI - Google Chrome
10:02 AM
BLAND UI
Nathan
localhost:8001/bland/instrument/

Welcome to BLAND UI

Scale Factor	Neutron Wavelength	u	v	w	Eta	Zero Position	2θ Min	2θ Max	Mode
1.5	1.912	.2	-.5	.4	0	0	10	155.4499969482422	D1A

24.1 KB

pbso4.dat

Remove file

Calculate

+ Add Phase

Phase 1 ×

Unit Cell

Space Group	a	b	c	Alpha	Beta	Gamma
Space group 62	8.478355407714844	5.396669387817383	6.9579691886901855	90	90	90

Atoms

Add Row

Label	Atom #	x	y	z	Occupancy	B
Pb	82 | Pb	0.18789000809192657	0.25	0.1671999990940094	0.5	0.5
S	16 | S	0.0645499974489212	0.25	0.6849799752235413	0.5	0.5
O1	8 | O	0.9086700081825256	0.25	0.5956499576568604	0.5	0.5
O2	8 | O	0.19308000802993774	0.25	0.5428600311279297	0.5	0.5
O3	8 | O	0.08072999864816666	0.027089998126029968	0.8084100484848022	1	0.5

0.5 KB

pbso4.cfl

Remove file

Calculate

BLAND UI - Google Chrome
10:12 AM
BLAND UI
Nathan
localhost:8001/bland/fit/

Welcome to BLAND UI

Fit all parameters

Check each parameter you want to fit.

Instrument

Scale Factor: 1.5 u: .2 v: -.5 w: .4 Eta: 0 Zero: 0

Phase 1

Unit Cell

a: 8.4784 b: 5.3967 c: 6.9580 Alpha: 90 Beta: 90 Gamma: 90

Atoms

Pb	x: 0.1879	y: 0.2500	z: 0.1672	Occupancy: 0.5	B Iso: 0.5
S	x: 0.0645	y: 0.2500	z: 0.6850	Occupancy: 0.5	B Iso: 0.5
O1	x: 0.9087	y: 0.2500	z: 0.5956	Occupancy: 0.5	B Iso: 0.5
O2	x: 0.1931	y: 0.2500	z: 0.5429	Occupancy: 0.5	B Iso: 0.5
O3	x: 0.0807	y: 0.0271	z: 0.8084	Occupancy: 1.0	B Iso: 0.5

localhost:8001/bland/fit/
residual_plot ....svg
plot (2).svg
SHOW ALL

localhost:8001/bland/status/2e46509e97/ - Google Chrome
10:26 AM
BLAND UI
localhost:8001/bla
Nathan
localhost:8001/bland/status/2e46509e97/

Fit Status

Complete!

Time: 694.616

O1 - occ: 0.453764604692908

O1 - x: 0.906028665942704

O1 - y: 0.226354683278558

O1 - z: 0.583637121295369

O2 - occ: 0.51395853014402

O2 - x: 0.189191978429432

O2 - y: 0.248505297659056

O2 - z: 0.550125050366528

O3 - occ: 1.0742436935351

O3 - x: 0.0825111838696875

O3 - y: 0.0209542888543863

O3 - z: 0.806063946955218

Pb - x: 0.192154072847831

Pb - y: 0.273718991943381

Pb - z: 0.166410419291456

S - occ: 0.49587772269697

S - x: 0.0589688051680264

S - y: 0.257644925232176

S - z: 0.669943652098611

b: 5.40582144528099

c: 6.96893397388366

scale: 1.2923300215113

params (12).json
residual_plot ....svg
plot (2).svg
SHOW ALL

Future Goals

- Model selection
- Instrument Control Integration

Acknowledgements

- NSF
- NIST
- NCNR
- CHRNS
- William Ratcliff
- Paul Kienzle
- Steve Disseler
- Julie Borchers
- Joe Dura
- Yamali Hernandez