diffcalc-core: Diffraction condition calculation package for a six-circle diffractometer

Irakli Sikharulidze

Diamond Light Source Ltd, Harwell Science and Innovation Campus, Didcot, Oxfordshire OX11 0DE, United Kingdom

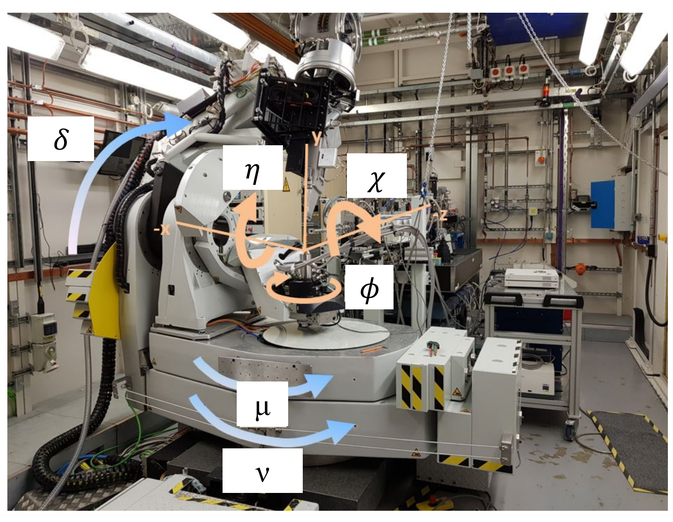
Introduction

diffcalc-core[1] is developed as a minimalist Python 3 API for calculating diffractometer positions for measuring radiation scattering in a specified reciprocal space directions.

The primary package feature if to provide a set of discreet solutions for diffractometer rotation angles that correspond to a specified sample reciprocal orientation.

The calculations are based on a formalism outlined for 4S+2D six-circle diffractometers in (You, H. 1999)[2].

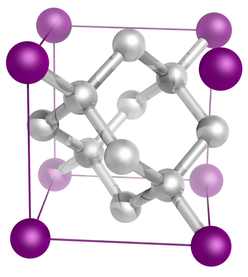
The code is derived from Diffcalc project that implemented reciprocal space calculations as a part of Generic Data Acquisition (GDA)[3] system used for beamline control.



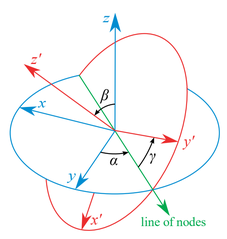
4S+2D six-circle diffractometer, I16 beamline, Diamond Light Source

Main diffcalc-core API workflow

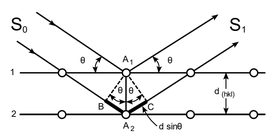
Setup sample orientation UB matrix.
- Create new *UBCalculation* object.
- Add unit cell parameters.
- Setup sample U rotation matrix.
  - Use two reference reflections/orientations.
  - Specify sample rotation angle and axis.
  - Trial matrix from a single reflection/orientation.



Set diffractometer operation mode using a combination of three angle/orientation constraints.
- Optional detector constraint.
- Optional reference orientation constraint.
- Required one, two or three sample orientation constraints.

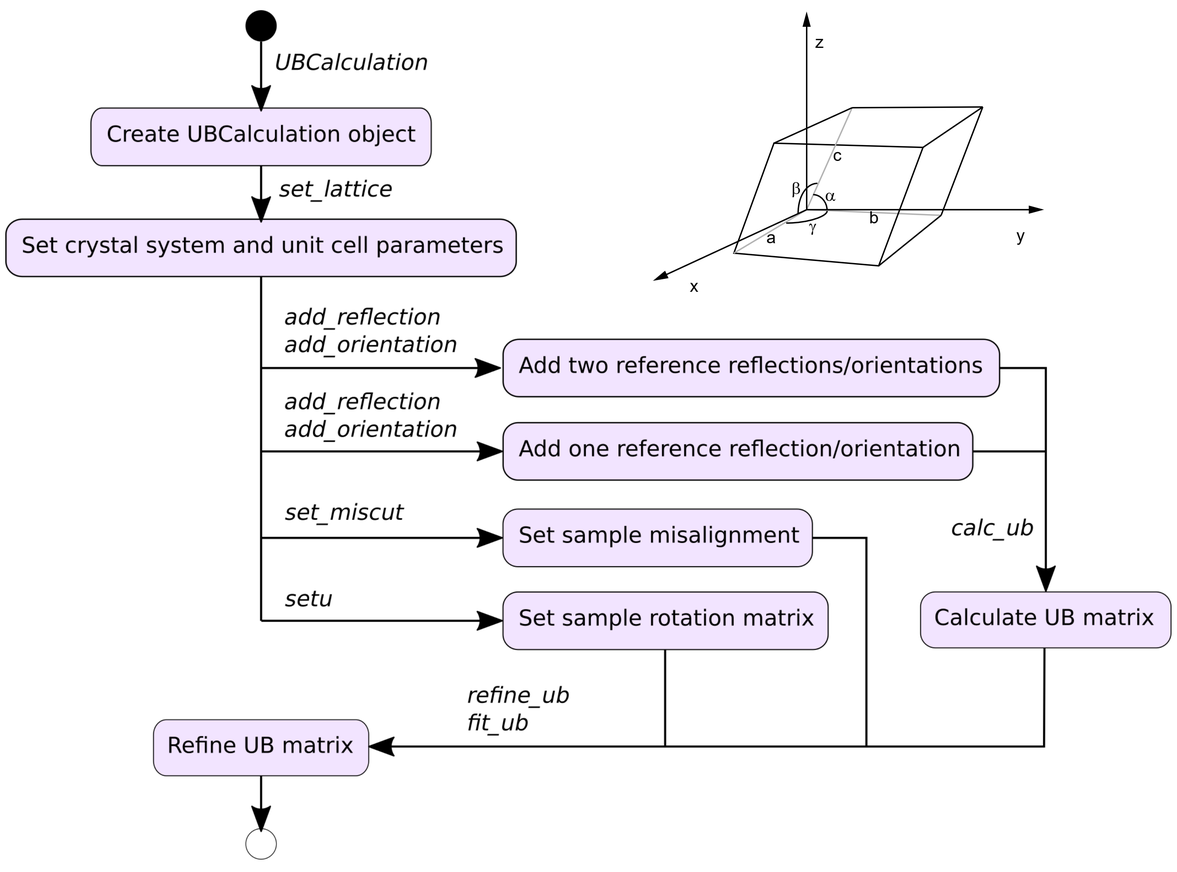


Get diffractometer position for a reflection of interest.
- Convert *(h, k, l)* reflection indices into diffractometer rotation angles or vice versa.
- Refine unit cell and sample orientation parameters using reflection data.
- Select alternative set of constraints in case of a degenerate scattering geometry.



UBCalculation: setting sample orientation

UBCalculation class provides methods for either setting sample orientation UB matrix or calculating it from a reflection/orientation data in various scenarios and further refining based on observed reflections.

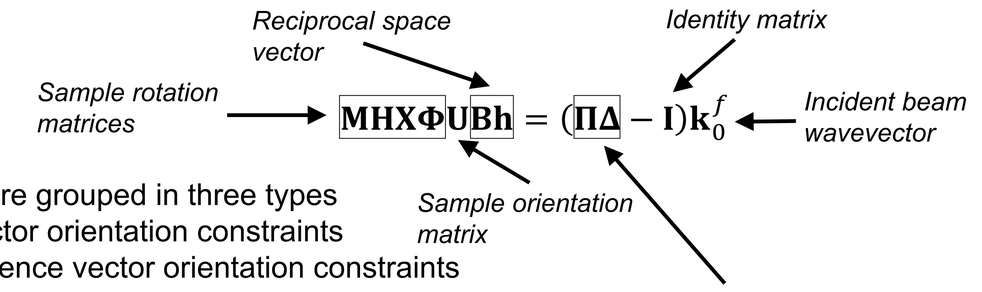


diffcalc-core API code example

```python
from diffcalc.hkl.calc import HklCalculation
from diffcalc.hkl.constraints import Constraints
from diffcalc.hkl.geometry import Position
from diffcalc.ub.calc import UBCalculation

# Create an object holding UB matrix information.
ubcalc = UBCalculation("sixcircle")

# Set Tetragonal unit cell parameters.
ubcalc.set_lattice("SiO2", 4.913, 5.405)

# Add reference reflection for UB matrix calculation.
# Set (0, 0, 1) reflection position to diffractometer
# angles  μ = 7.31° and ν = 10.62° at 12.39842 keV.
ubcalc.add_reflection(
    (0, 0, 1),
    Position(7.31, 0.0, 10.62, 0, 0.0, 0.0),
    12.39842,
    "refl1",
)
# Add reference crystal orientation (0, 1, 0)
# pointing in (0, 1, 0) direction in a laboratory frame.
ubcalc.add_orientation((0, 1, 0), (0, 1, 0), None, "plane")

# Calculate UB matrix from the above references.
ubcalc.calc_ub("refl1", "plane")

# Set diffractometer operation mode.
# qaz   - constrain scattering plane azimuthal orientation.
# alpha - constrain reference vector orintation.
# eta   - constrain diffractometer η angle.
cons = Constraints({"qaz": 0, "alpha": 0, "eta": 0})

# Create an object for reciprocal space calculation
hklcalc = HklCalculation(ubcalc, cons)

# Get all diffractometer orientations for (0, 0, 1)
# reflection at 1.0 Å wavelength.
hkl = (0, 0, 1)
wavelength = 1.0
all_pos = hklcalc.get_position(*hkl, wavelength)

# Get (h, k, l) reflection indices for
# diffractometer position μ = 7.31° and ν = 10.62°
# at 1.0Å wavelength.
pos = Position(7.31, 0.0, 10.62, 0.0, 0.0, 0.0)
hkl = hklcalc.get_hkl(pos, wavelength)
```

Constraints: diffractometer operation modes

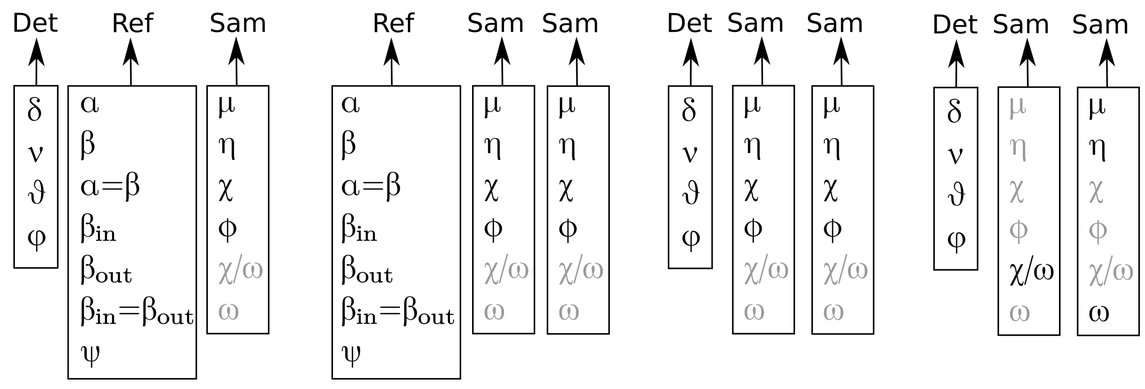
Three constraints for detector and/or sample orientation are necessary to obtain a set of discrete solutions from the diffractometer equation.



Constraints are grouped in three types
- Detector orientation constraints
- Reference vector orientation constraints
- Sample orientation constraints

Implemented constraint combination classes
- 1x Detector; 1x Reference; 1x Sample
- 1x Reference; 2x Sample
- 1x Detector; 2x Sample

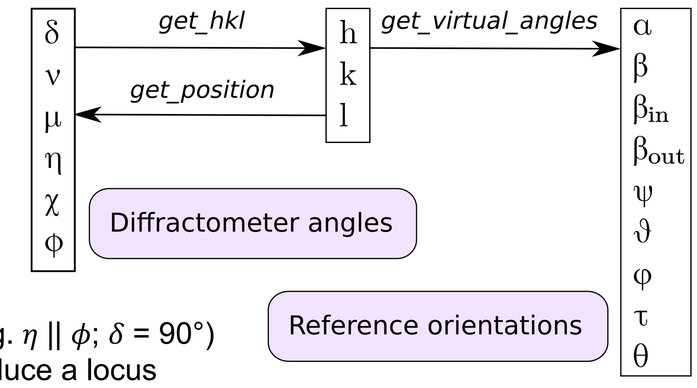
All sets of implemented constraint combinations can be obtained by selecting one angle from each column excluding grayed-out elements



HKLCalculation: reciprocal space calculations

HKLCalculation class provides methods for converting between diffractometer angles and reciprocal space *(h, k, l)* orientations based on sample and diffractometer operation information *UBCalculation* and *Constraints* objects.

In fully constrained diffractometer geometry, each *(h, k, l)* value corresponds to a discreet set of diffractometer angles with a corresponding set of reference vector and scattering plane orientations.



In some degenerate geometries (e.g. $\eta \parallel \phi$; $\delta = 90°$) complete set of constraints can produce a locus of solutions that raises an exception indicating that the selected constraints are inadequate to produce a discreet set of diffractometer angles.

Instruments currently supported at Diamond Light Source
- I16: Materials and Magnetism[4] (4S+2D kappa diffractometer)
- I07: Surface and interface diffraction[5] (2S+2D surface diffractometers)
- I21: Resonant Inelastic X-ray Scattering[6] (3S+1D RIXS spectrometer)

References

[1] https://github.com/DiamondLightSource/diffcalc-core/
[2] You, H., *Angle calculations for a '4S+2D' six-circle diffractometer.* J. Appl. Cryst. (1999) **32**, 614-623.
[3] http://www.opengda.org/
[4] Collins, S. P., et. al. *Diamond Beamline I16 (Materials & Magnetism)* AIP Conf. Proc. (2010) **1234**, 303–306.
[5] Nicklin, C., et. al. *Diamond beamline I07: a beamline for surface and interface diffraction*. J. Synchrotron Rad. (2016) **23**, 1245–1253.
[6] https://www.diamond.ac.uk/Instruments/Magnetic-Materials/I21.html

Acknowledgements

Rob Walton (author Diffcalc for GDA), Rose Yemelyanova, Joe Shannon, Fajin Yuan (GDA), Mirian Garcia-Fernandez, Ke-Jin Zhou (I21), Gareth Nisbet, Dan Porter, Steve Collins, Alessandro Bombardi (I16), Jonathan Rawle, Francesco Carla (I07)

Science & Technology Facilities Council

wellcome trust

For more information please visit https://diffcalc-core.readthedocs.io
or contact *Irakli Sikharulidze* at irakli.sikharulidze@diamond.ac.uk

diamond