Processing scattering and diffraction data

Adventures in k-space, part 2

Lenson Pellouchoud
SLAC / SSRL XRSD summer school
7/17/2018

lenson@slac.stanford.edu
Outline

Calibration and remeshing
- Solving the experiment geometry
- Mapping measurements to q-space
- Grazing incidence (GI)

Cleaning
- Background subtraction
- Cosmic rays

Fitting / refinement
- Scattering and structural parameters
- Fit objectives
Scattering measurements
real space probe → reciprocal space information

Incident light:
Approximate plane-wave E-field

\[ E(\mathbf{r}, t) = E_0 \exp(i(\omega t - \mathbf{k} \cdot \mathbf{r})) \]

Scattering
• All parts of the scatterer distribution \( \rho(\mathbf{r}) \) contribute to the scattered field
• A detector at a certain location (in real space) gives information about the charge distribution (in real space)
• All of the intermediate analysis occurs in reciprocal momentum transfer (\( \mathbf{q} \)) space

Distribution of scatterers
- Single scatterer
- Multiple scatterers
- Same distribution, different wavelength

Position of measurement
Scattered intensity
What do we get from the detector?

- Detector position selects a \( q \) value
- Detector has finite size: a given position integrates a small, finite range of \( q \)
- Detector has finite speed: the measured intensity is integrated over time
- We measure the “integrated intensity” pattern \( I(q) \), proportional to the time-averaged square of \( E(q) \)
- Exact (non-integrated) intensity profiles are not generally feasible to measure
Area detectors

- Most modern experiments employ area detectors
- Essentially, thousands or millions of detectors operating simultaneously in parallel
- Each pixel selects a different range of $q$ values
- Each pixel requires different geometric corrections
- Analysis is not easier, but the experiment happens faster and the $q$-range is measured concurrently
Scattering analysis 1: calibration
the experiment geometry (credit: pyFAI)

- 3 distances in meter: \( \text{dist}, \text{poni}_1, \text{poni}_2 \)
- 3 rotation in radians: \( \text{rot}_1, \text{rot}_2, \text{rot}_3 \)
- wavelength / energy

From the sample’s point of view,
Looking at the detector:

\( \text{rot}_1 \uparrow \): move detector to the right
\( \text{rot}_2 \uparrow \): move detector downwards
\( \text{rot}_3 \uparrow \): move detector clockwise

https://github.com/silx-kit/pyFAI
Scattering analysis 1: calibration
solving the experiment geometry (credit: pyFAI)

- Measure a well-known material
- Initial condition: estimate the experimental geometry
- Refine geometry until the computed pattern matches the measurement
- Once solved, each pixel can be assigned a $q$ range

Software packages for calibration:
- pyFAI
- scikit-beam
- Xi-CAM
- Nika (Igor)
- Several more, none comprehensive

https://github.com/silx-kit/pyFAI
Scattering analysis 1: calibration
mapping real space measurements to q-space

- After calibration, each detector position (or pixel) is assigned to a range of \( q \) vectors

- Measured patterns in **pixel** space can now be mapped onto bins covering a plane in \( q \) which can be summed into bins in \( q \)

- Point detectors: the \( q \)-range is directly related to the detector position – remeshing is unnecessary
Scattering analysis 1: calibration
mapping real space measurements to q-space

- Raw CCD SAXS output
- Looks like form factor scattering (see no rings)
- Cannot interpret the form factor without q-space remeshing
Scattering analysis 1: calibration
mapping real space measurements to q-space

- Re-meshed and integrated into bins of $q, \chi$
- For isotropic samples, this is more useful than $q_x, q_y$ remeshing
- Can now interpret the scattering
Scattering analysis 1: calibration
mapping real space measurements to q-space

- Re-meshed and integrated to $q$-bins

- Can now subtract background (integrated flat field) and fit scattering pattern to isotropic form factors in $q$ (more on this ahead)
Scattering analysis 1: calibration grazing incidence (credit: pygix)

- For a planar (or fibre) sample, $\mathbf{q}$-vectors with non-normal components are probed wherever $\alpha \neq \beta$
- The vertical line on the detector image does not represent the $q_z$-line
- GI remeshing splits these central bins to their true $\mathbf{q}$-regions

https://github.com/tgdane/pygix
Scattering analysis 2: corrections

• An active detector element can be triggered by a cosmic ray

• Rare events: the vast majority can be removed by taking measurements in pairs (some detector controls will do this automatically)

• Else, they can be found during post-processing and removed or interpolated

\[ I(q) \]

zinger

signal

q
Scattering analysis 2: corrections on cosmic rays

- Shown right: zinger found during post-processing

- Simple heuristic: compare intensity to mean value of nearby intensities on either side, set a threshold

- Interpolation: replace zinger with mean value of nearby intensities, excluding flagged zingers
Scattering analysis 2: corrections
background subtraction

- dark field: detector readout with no light
- sometimes this is subtracted by default on the detector control PC: dark field is held in memory

- flat field: detector readout with everything but the sample
- if flat field is the entire background, it can be subtracted directly
- else, background is approximated, ideally on some physical basis
Scattering analysis 2: corrections
background subtraction

- Strong signal: data could be interpreted without subtraction, but subtraction brings out lower-intensity features

- Weak signal: data are unrecognizable before background subtraction
Scattering analysis 3: fitting and refinement
form factor and structural parameters

- Scattered intensity: magnitude squared of the scattered plane waves of all scatterers
  \[
  E(q) = E_0 \frac{r_e}{R} \exp(i(\omega t - kR)) \sum_p [f(q) \exp(iq \cdot R_p)]
  \]
  \[
  I(q) = |E(q)|^2 = E'(q) E(q)
  \]

- Equivalently: sum over all scattering populations of their characteristic intensity equations
  \[
  I(q) = c \epsilon_0 \left( \frac{E_0 r_e}{R} \right)^2 \sum_{p \in \text{dilute}} f_p(q)^2
  \]
  \[
  + c \epsilon_0 \left( \frac{E_0 r_e}{R} \right)^2 \sum_{r \in \text{condensed}} f_r(q)^2 S_r(q)
  \]
  \[
  + c \epsilon_0 \left( \frac{E_0 r_e}{R} \right)^2 \sum_{s \in \text{crystals}} F_s(q)^2 \sum_{h,k,l} P_{s,hkl}(q)
  \]

Fit form factors with respect to scatterer shape parameters (e.g. sphere radius)

Fit structure factors with respect to population parameters (e.g. volume fraction, interparticle potential)

Fit broadening function (easy) or disorder parameters (hard)

Fit form factors and scatterer positions within unit cell

Fit unit cell parameters
Scattering analysis 3: fitting and refinement objective analysis

- Goal: remove human judgment from fitting problems
- Direct least-squares
  - Sensitive to noise
  - Preferentially fits larger values
- Uncertainty-weighted least squares
  - Less sensitive to noise
- Log-weighted least squares
  - More sensitive to noise
  - Treats large and small values more equally
  - Can be uncertainty-weighted as well

\[
\chi^2 \equiv \frac{1}{\Omega} \int d^3q \left[ I(q) - I_{\text{meas}}(q) \right]^2
\]

\[
\text{argmin}_{\text{params}} \left[ \chi^2(\text{params}) \right]
\]

\[
\sigma^2 \equiv \frac{1}{\Omega} \int d^3q \left[ \sigma_I(q) \left( I(q) - I_{\text{meas}}(q) \right) \right]^2
\]

\[
\text{argmin}_{\text{params}} \left[ \sigma^2(\text{params}) \right]
\]

\[
\chi_{\log}^2 \equiv \frac{1}{\Omega} \int d^3q \left[ \log \left( I(q) \right) - \log \left( I_{\text{meas}}(q) \right) \right]^2
\]

\[
\text{argmin}_{\text{params}} \left[ \chi_{\log}^2(\text{params}) \right]
\]
Objective:

- Logarithmic
- Uncertainty-weighted

7 Parameters:

- Particle (spherical):
  - $r_0$ (mean radius)
  - $\sigma$ (std of radius)
  - $I_0$ (scaling)

- Precursor (Guinier-Porod)
  - $r_g$ (radius of gyration)
  - $D$ (porod exponent)
  - $G$ (scaling)

- Noise
  - $I_0$ (flat noise floor)
The end – thank you!

Special thanks to the Tassone group at SSRL!
Chris Tassone, Amanda Fournier, Liheng Wu, Anthony Fong,
Karsten Bruening, Tim Dunn, Ekaterina Tcareva

… and to the XRSD summer school organizing committee:
Apurva Mehta, Kevin Stone, Chris Tassone, Mike Toney

Reference materials:

- **Remeshing and integration:**
  pyFAI (python Fast Azimuthal Integration)
  https://github.com/silx-kit/pyFAI

- **Grazing incidence:**
  pygix (python grazing-incidence xrd)
  https://github.com/tgdane/pygix

- **Data processing and plots:**
  paws (the Platform for Automated Workflows by SSRL)
  https://github.com/slaclab/paws