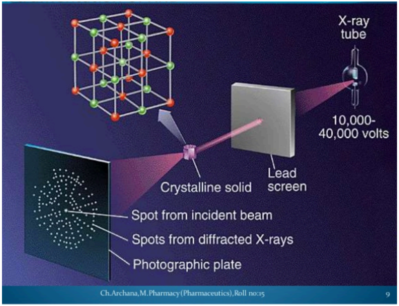
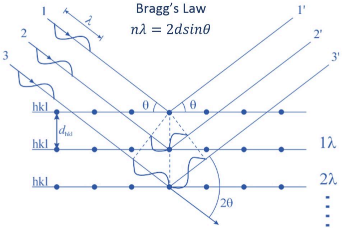
NYSDS 2016, New York, NY August 14-17, 2016

DiffPy-CMI – an extensible software framework for multimodal analysis of atomic structure of nanomaterials

Pavol Juhás, Hubertus Van Dam,
Simon J. L. Billinge

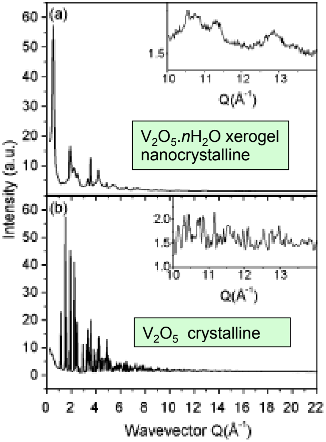
Atomic structure of materials



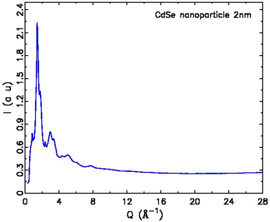


- atomic structure is fundamental for characterization and understanding of materials
- most structures have been determined from X-ray diffraction experiments
- for crystal structures the scattering pattern reduces to several tens / hundreds of Bragg reflections (corresponding to planes of atoms in the crystal)
- structure can be depicted by a few variables (unit cell parameters, positions of symmetry-independent atoms) → over-constrained problem → structure determination is routine

Nanostructure problem



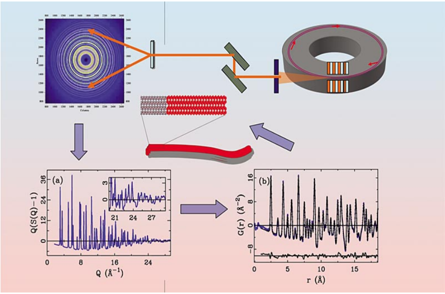
[V. Petkov, et. al., *J. Am. Chem. Soc.* **121**, 10157 (2002)]

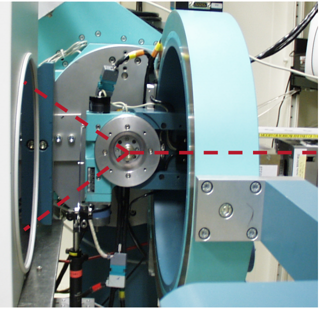


- none or few Bragg reflections → conventional crystallography fails
- assumption of crystal periodicity → local distortions may be missed by conventional crystallography
- need to use other probes for nanoscale structures

Total scattering PDF technique

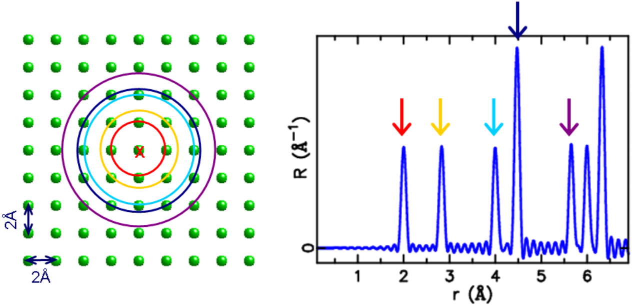
- use the entire diffraction pattern → Bragg peaks AND the diffuse scattering
- Fourier transformation of TS data gives Pair Distribution Function → direct probe of interatomic distances





- Rapid Acquisition PDF experiment,
  Chupas et al., J. Appl. Crystallogr. (2003)
- 100 ms exposure times, can handle in-situ studies

PDF – the atomic Pair Distribution Function



Pair distribution function (PDF) provides probability of finding distance “*r*” between two atoms in the material.

Methods for analyzing experimental PDFs

direct readout

- bond distances and their variations

big box modeling

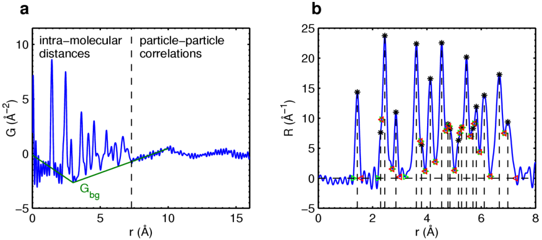
- Reverse Monte Carlo (RMC) [Pusztai & McGreevy, Physica B, 234-236, (1997)]
- ~10^4 atoms in a large box with periodic boundary conditions
- MC position optimization → excellent fit to the experimental PDF
- many degrees of freedom – RMC modeling requires constraints to produce physically feasible structures
- interpretation of 10^4 coordinates – bond length and angle statistics

small box modeling

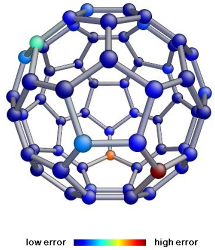
- up to ~100 atoms in a small cell with periodic boundary conditions
- PDF modeling can be focused to a short, specific length scale
- **simple refinement**
  - start with reasonably accurate initial structure
  - downhill minimization of the model variables to fit observed PDF
- **structure determination**
  - extract experimental pair-distances from the PDF
  - find shape that reproduces the same set of pair distances
- **complex refinement**
  - additional cost terms (atom overlap, bond valence sums,…)
  - mixed contributions from molecules and crystalline phases

Successful structure determination from PDF

- PDF data from neutron diffraction of C_{60}
- convert PDF data to a list of atom distances (60 atoms, 1770 distances)
- extracted 18 out of 21 unique distance values
- structure determination was still successful



[Juhás et. al, *Nature* 440, 655-658 (2006)]
[Juhás et. al, *Acta Cryst. A* 64, 631-640 (2008)]



- highly symmetric rigid molecule → sharp, well resolved PDF peaks → measured signal carries enough information to solve the structure

Journal of
Applied Crystallography
ISSN 0021-8898

Received 20 October 2009
Accepted 16 March 2010

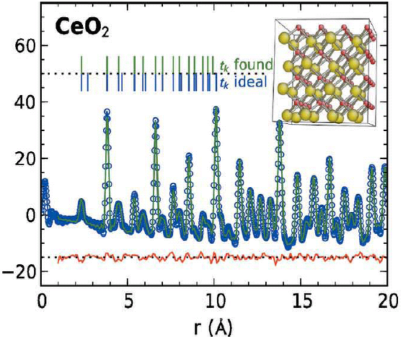
Crystal structure solution from experimentally determined atomic pair distribution functions

P. Juhás,[a]* L. Granlund,[b] S. R. Gujarathi,[b] P. M. Duxbury[b] and S. J. L. Billinge[a,c]

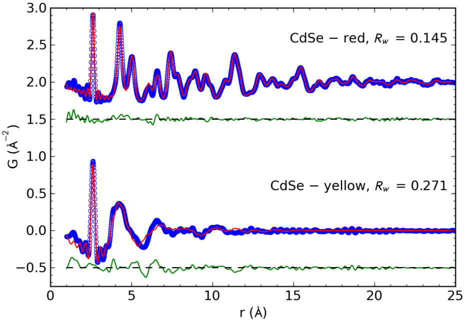
[a]Department of Applied Physics and Applied Mathematics, Columbia University, New York, NY 10027, USA, [b]Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA, and [c]Condensed Matter Physics and Materials Science Department, Brookhaven

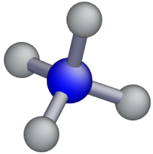
C_d and C_c are the distance and atom-overlap costs, as defined in equations (3) and (4). s_x, s_y and s_z are the standard deviations in the f normalized to a simple [111] cell. s_r (Å) is the root mean-square displacement of the solved sites from the reference CIF positions.

Sample (supercell)	Atoms	Cost C_d (0.01 Å^2)		Cost C_c (Å^2)		Deviation of coordinates	
		Liga	CIF	Liga	CIF	s_x	s_y
Successful solutions							
Ag [111]	4	0.0232	0.136	0	0.001	0	0
Ag [222]	32	0.0097	0.136	0	0.001	0.00025	0.00024
$BaTiO_3$ [111]	5	0.370	0.394	0.040	0.042	0.0057	0.0066
$BaTiO_3$ [112]	10	0.392	0.394	0.058	0.042	0.00023	0.039
C graphite [111]	4	0.396	0.574	0.010	0.016	0.0029	0.0029
C graphite [221]	16	0.420	0.574	0.010	0.016	0.0086	0.0065
CdSe [111]	4	0.107	0.138	0	0.001	0	0
CdSe [221]	16	0.0856	0.138	0	0.001	0.00010	0.00013
CeO_2 [111]	12	0.515	0.554	0	0	0	0
NaCl [111]	8	1.75	1.71	0	0	0	0
NaCl [222]	64	1.20	1.71	0	0	0.00031	0.00031
Ni [111]	4	0.0024	0.0024	0	0	0	0
Ni [222]	32	0.0025	0.0024	0	0	0.00015	0.00013
PbS [111]	8	0.0125	0.0104	0.010	0.011	0	0
PbS [222]	64	0.0140	0.0104	0.010	0.011	0.00005	0.00004
PbTe [111]	8	0.0024	0.0127	0.097	0.090	0	0
PbTe [222]	64	0.0022	0.0127	0.097	0.090	0.00011	0.00011
Si [111]	8	0.0045	0.0045	0	0	0	0
Si [222]	64	0.0048	0.0045	0	0	0.00010	0.00009
$SrTiO_3$ [111]	5	0.437	0.437	0.002	0.002	0	0
Zn [111]	2	0.495	0.470	0	0	0	0
Zn [222]	16	0.564	0.470	0	0	0.00010	0.00006
ZnS sphalerite [111]	8	0.150	0.0647	0	0	0	0
ZnS sphalerite [222]	64	0.160	0.0647	0	0	0.00029	0.00033
ZnS wurtzite [111]	4	0.141	0.152	0	0	0	0
ZnS wurtzite [221]	16	0.165	0.152	0	0	0.00003	0.00002
Failed solutions							
$CaTiO_3$ [111]	20	0.4967	0.902	0.52	0.072	0.16	0.14
TiO_2 rutile [111]	6	0.5358	0.758	0.40	0.009	0.081	0.24



Failed structure determination from PDF

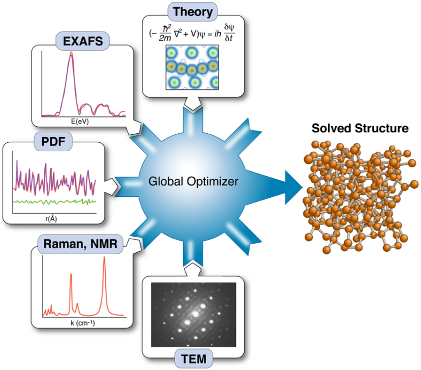




- CdSe nanoparticles from Dr. Cossairt, Chemistry Dept., Columbia Univ.
- X-ray PDF measured at X17B, NSLS, Brookhaven laboratory
- **red-phase** well modeled by a mix of bulk CdSe phases
- **yellow-phase** ($Cd_{35}Se_{28}$) has more complicated structure (distorted A-B_4 tetrahedra), poorly resolved peaks → no luck with structure solution

Cossairt, B. et al., J. Phys. Chem. Lett. 2, 3075−3080 (2011)

Complex modeling



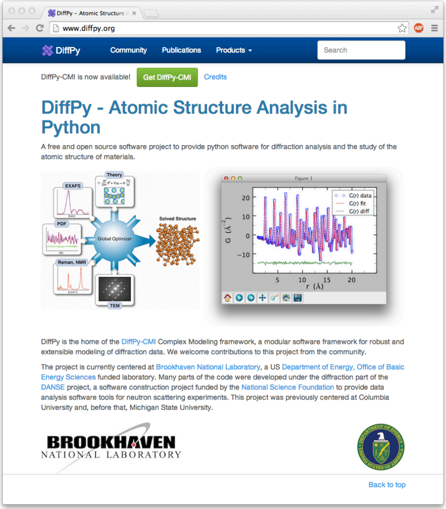
Problem

- not enough information in the available experimental data

Remedy

- collect data from multiple experimental techniques
- use additional knowledge about the studied material - chemical constraints, rigid units, bond-valence sums, energy calculation
- combine all experimental and theoretical inputs about the structure in one optimization scheme
- requires flexible software tools to setup custom models adaptable for specifics of studied materials.

DiffPy-CMI – Complex Modeling Infrastructure



- tools for PDF, BVS, SAS simulations, structure data handling, multi-input optimizations
- Python and C++, object-oriented, reusable, extensible libraries
- code-base derives from DANSE, **http://danse.us/** Caltech, SNS/ORNL

DANSE

- available since March 2014, **http://www.diffpy.org/** for Linux, Mac, UNIX systems

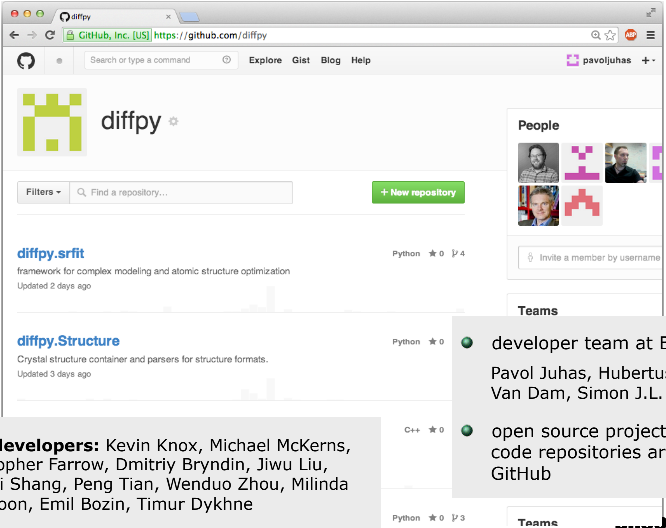
upgrade release on March 2016

- available for Anaconda Python on Linux and Mac:

$ conda install –c diffpy diffpy-cmi

BROOKHAVEN
NATIONAL LABORATORY

DiffPy-CMI – open source project

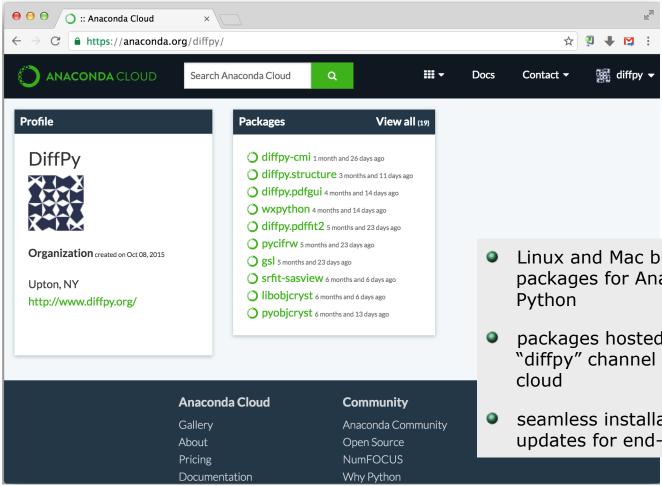


- developer team at BNL:

  Pavol Juhas, Hubertus Van Dam, Simon J.L. Billinge
- open source project, all code repositories are on GitHub

past developers: Kevin Knox, Michael McKerns, Christopher Farrow, Dmitriy Bryndin, Jiwu Liu, Yingrui Shang, Peng Tian, Wenduo Zhou, Milinda Abeykoon, Emil Bozin, Timur Dykhne

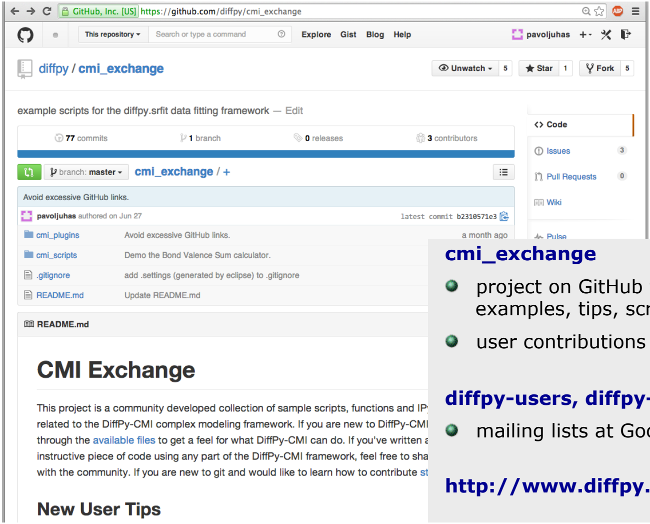
DiffPy-CMI – software deployment



- Linux and Mac binary packages for Anaconda Python
- packages hosted in the "diffpy" channel at Anaconda cloud
- seamless installation and updates for end-users

BROOKHAVEN NATIONAL LABORATORY

DiffPy-CMI – user community support



cmi_exchange

- project on GitHub to provide examples, tips, script sharing
- user contributions welcome

diffpy-users, diffpy-dev

- mailing lists at Google Groups

http://www.diffpy.org/

DiffPy-CMI overview

- Complex Modeling Infrastructure – a software toolbox for multi-probe structure analysis
- collection of Python and C++ libraries responsible for tasks needed in structure analysis (structure representation, forward calculators, refinement configuration)
- object oriented architecture. designed for extensibility, code reuse, support for integration with other crystallographic packages.
- computationally intensive parts coded in C++, designed for speed and extensibility. Can be used as a pure C++ library "libdiffpy".
- C++ objects are exposed to Python using boost python library. Derived classes can be defined in C++ or in Python and then used from either language.
- Calculator are composed from objects responsible for partial tasks. These objects can be configured, tweaked or replaced at runtime.
- no GUI, simulations and structure refinements are configured from Python scripts.

DiffPy-CMI – functionality overview

Structure Representation

- **diffpy.Structure** → simple storage of P1 periodic structures, finite clusters, input and ouput for CIF, PDB, xyz, pdffit, discus formats. Space group definitions, symmetry expansion, generation of symmetry-based constraints.
- **pyobjcryst** → advanced structure representations, crystals with space group, crystals containing rigid molecules, bond-length and bond-angle restraints, z-matrix representation. Input and output in custom XML and CIF formats. Python interface to the ObjCryst++ crystallographic library by V. Favre-Nicolin, [J. Appl. Cryst. 35 (2002), 734-743]

Forward Calculators

- **diffpy.srreal** → calculators of structure-based physical quantities, such as PDF, Debye sum, bond lengths, bond valence sums, overlap of empirical atom radii.
- **pyobjcryst** → powder and single-crystal diffraction patterns
- **srfit-sasview** → selected functions for Small Angle Scattering simulations from the SasView program, http://www.sasview.org

Fit configuration and management

- **diffpy.srfit** → setup and control of general fitting problems, control of constraints and restraints, setup of refinements to multiple data sources, simple analysis of fit results

C++ libraries

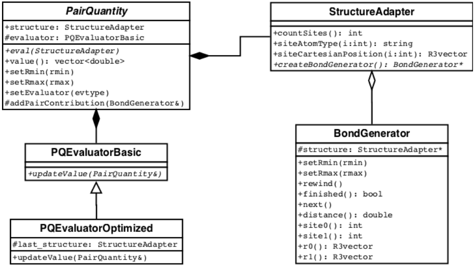
- **libdiffpy** – computationally expensive parts - PDF, BVS, etc. Calculation of pair-sum based quantities.
- **libObjCryst** – free objects for crystallography by Vincent Favre-Nicolin, [J. Appl. Cryst. 35 (2002), 734-743].

BROOKHAVEN NATIONAL LABORATORY

PairQuantity – a template calculator

- the base calculator – abstract recipe for evaluating physical quantities derived from pair-interactions.

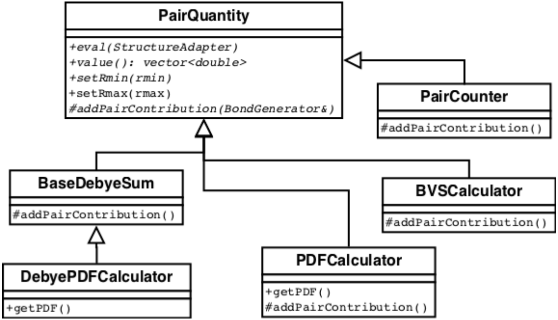
$$P(r_1, r_2, \ldots, r_N) = \sum_{i,j}^{N} p(r_{ij})$$



- common structure adapters and distance generation code
- support for partial sums
- option for parallel evaluation
- support for fast updates by re-evaluating contributions from the changed atoms

PairQuantity-derived calculators

- derived concrete calculators override the *addPairContribution()* method



BondCalculator

- calculate oriented bond vectors up to a specified distance limit
- optional filtering by atom types, site indices, direction cones

example:

```
>>> from pyobjcryst.crystal import CreateCrystalFromCIF
>>> from diffpy.srreal.bondcalculator import BondCalculator
>>> rutile = CreateCrystalFromCIF(open('TiO2_rutile.cif'))
>>> bc = BondCalculator(rmax=2)
>>> bc(rutile)
array([ 1.94720295,  1.94720295,  1.94720295,  1.94720295,  1.94720295,
        1.94720295,  1.98177183,  1.98177183,  1.98177183])
>>> for i in zip(bc.distances, bc.types0, bc.types1, bc.directions):
...     print i
...
(1.9472029472402153, 'Ti', 'O', array([-0.8951757,  0.8951757, -1.4795]))
(1.9472029472402153, 'Ti', 'O', array([-0.8951757,  0.8951757,  1.4795]))
(1.9472029472402153, 'Ti', 'O', array([ 0.8951757, -0.8951757, -1.4795]))
(1.9472029472402153, 'Ti', 'O', array([ 0.8951757, -0.8951757,  1.4795]))
(1.9472029472402153, 'O', 'Ti', array([ 0.8951757,  0.8951757, -1.4795]))
(1.9472029472402153, 'O', 'Ti', array([ 0.8951757,  0.8951757,  1.4795]))
(1.9817718303429834, 'Ti', 'O', array([-1.4013243, -1.4013243,  0.     ]))
(1.9817718303429837, 'Ti', 'O', array([ 1.4013243,  1.4013243,  0.     ]))
(1.9817718303429837, 'O', 'Ti', array([-1.4013243, -1.4013243,  0.     ]))
```

BROOKHAVEN NATIONAL LABORATORY

BVSCalculator

- bond valence sums – approximate formula for ion valences

Brese, Acta Cryst. B47, 192-197 (1991)

$$
\begin{aligned}
v_{ij} &= \exp\left[\frac{R_{ij} - d_{ij}}{b}\right] \\
V_i &= \sum_j v_{ij}
\end{aligned}
$$

- evaluates valence at each site, BVS difference, mean square BVS difference which accounts for partial occupancies and site multiplicities
- related: class BVParametersTable
  - lookup of bond valence parameters, [bvparm2009.cif by I. D. Brown]
  - option to define and revert custom BVS parameters

example:

```
>>> from pyobjcryst.crystal import CreateCrystalFromCIF
>>> from diffpy.srreal.bvscalculator import BVSCalculator
>>> sto = CreateCrystalFromCIF(open('SrTiO3.cif'))
>>> bvsc = BVSCalculator()
>>> bvsc(sto)
array([ 2.12652479,  4.16096701, -2.0958306 ])
>>> bvsc.bvdiff
array([-0.12652479, -0.16096701, -0.0958306 ])
>>> bvsc.bvmsdiff
0.013893882037591496
```

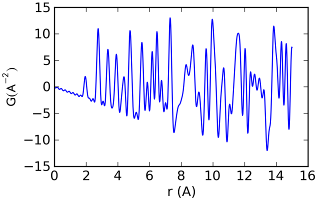
BROOKHAVEN NATIONAL LABORATORY

PDFCalculator

- PDF calculation in real-space
  - suitable for periodic systems
  - one structure per calculator → mixed-phase PDFs obtained by combining several PDFCalculator objects
- other results: radial distribution function, partial PDFs, $F(Q)$
- class ScatteringFactorTable
  - lookup of xray, netron or electron scattering factors
  - support for custom scattering factors
- class PeakProfile – the profile function for a pair contribution
- class PeakWidthModel – calculates profile width for a given atom pair
- class PDFEnvelope – one or more r-dependent scaling envelopes
- class PDFBaseline – the baseline function, by default $-4\pi\rho_0 r$

example:

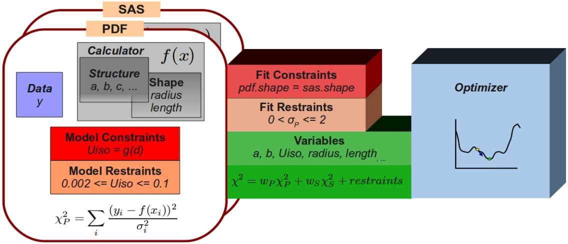
```
>>> from diffpy.Structure import Structure
>>> from diffpy.srreal.pdfcalculator import PDFCalculator
>>> sto = Structure(filename='SrTiO3.cif')
>>> pdfc = PDFCalculator(rmax=15, qmax=25)
>>> r, g = pdfc(sto)
>>> import pylab
>>> pylab.plot(r, g)
```



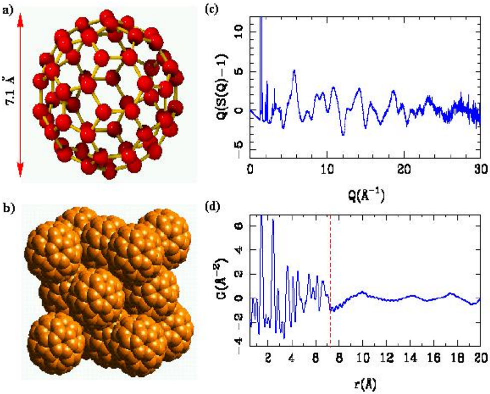
BROOKHAVEN
NATIONAL LABORATORY

SrFit – multi-component fit manager

- Python module for general multi-component data refinement
- construct FitContribution by associating observed data with simulation
  - models can be defined with built-in calculators, math expressions, Python functions
  - model parameters are exposed to SrFit. Parameters can be constrained or restrained, e.g., "*a* = *b* = *c*" for cubic structure
- FitContributions are combined to a single total cost function (residual vector or scalar value) with interface suitable for optimization routines
- control functions to fix/free variables, define constraints, restraints, hook functions
- post-processing to generate fit result reports – partial costs per each contribution, error estimates and correlations of the fit variables.



PDF modeling of fcc-C_{60}

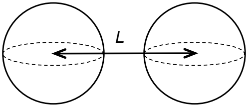


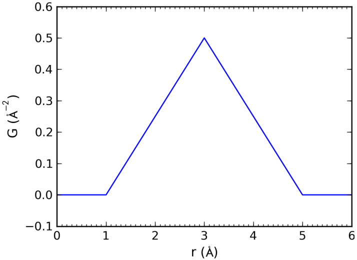
- neutron PDF measured on C_{60} fcc structure [GLAD IPNS, E. Bozin]
- low-*r* sharp peaks – correlations within C_{60}
- high-*r* broad peaks – correlations between randomly oriented balls

Can we simulate PDF on a full measured range?

- calculate as a sum of single particle PDF and PDF from a lattice of spherical shells

PDF peak profile for spherical shells





- PDF from two spherical shells can be calculated analytically

$$G(r) = \frac{1}{S_1 S_2 r} \iint_{S_1 S_2} \delta(r - r_{12})\, \mathrm{d}S_1 \mathrm{d}S_2$$

triangular profile centered at sphere separation *L* and with FWHM equal *D*

- *fcc* arrangement of spherical shells → PDF calculation requires <u>triangular profile function</u>

DiffPy-CMI supports user-defined profiles for PDF simulations.

BROOKHAVEN NATIONAL LABORATORY

Definition of custom peak profile

profile function defined in C++

```
#include <cmath>
#include <diffpy/srreal/PeakProfile.hpp>

using diffpy::srreal::PeakProfile;
using diffpy::srreal::PeakProfilePtr;

class SphericalShellsProfile : public PeakProfile {
public:
    PeakProfilePtr create() const {
        return PeakProfilePtr(new SphericalShellsProfile());
    }

    PeakProfilePtr clone() const {
        return PeakProfilePtr(new SphericalShellsProfile(*this));
    }

    const std::string& type() const {
        static std::string tp = "sphericalshells-cpp";
        return tp;
    }

    double yvalue(double x, double fwhm) const
    {
        if (fabs(x) > fwhm)  return 0.0;
        double rv = (fwhm - fabs(x)) / (1.0 * fwhm * fwhm);
        return rv;
    }

    double xboundlo(double fwhm) const   { return -fwhm; }

    double xboundhi(double fwhm) const   { return +fwhm; }
};

bool reg_SawToothProfile = SphericalShellsProfile().registerThisType();
```

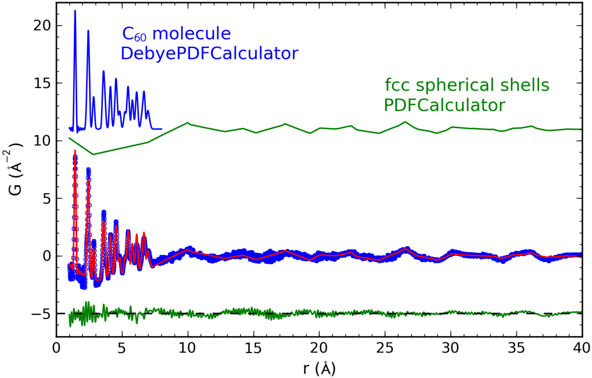
PDFCalculator configured in Python to use new profile

```
>>> from diffpy.srreal.pdfcalculator import PeakProfile, PDFCalculator
>>> PeakProfile.getRegisteredTypes()
set(['croppedgaussian', 'gaussian'])
>>> import ctypes
>>> ctypes.cdll.LoadLibrary('./sphericalshells-cpp.so')
>>> PeakProfile.getRegisteredTypes()
set(['sphericalshells-cpp', 'croppedgaussian', 'gaussian'])
>>> pdfcalc = PDFCalculator()
>>> pdfcalc.setPeakProfileByType('sphericalshells-cpp')
```

- new profile functions can be added either in Python or C++
- for C++ the profile function is compiled as a dynamic link library sphericalshells-cpp.so
- when loaded the library adds new profile to the global registry → profile is ready for use in Python
- no need to rebuild any C++ library in DiffPy-CMI
- no need to write Python wrappers for the new profile function

BROOKHAVEN NATIONAL LABORATORY

PDF refinement of fcc C_{60}

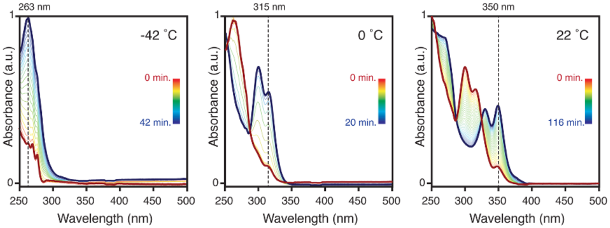


fit residuum R_w = 0.26
molecule diameter = 7.113(2)
scale ratio = 60.0(1)
shell diameter = 7.22(4)
U_{iso} = 0.00323(4)

- PDF from fcc C_{60} can be refined on the full measured range accounting for both intra and inter-molecular correlations

Structure of CdSe quantum dots

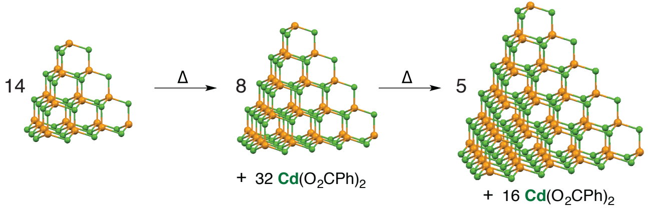
- collaboration with Prof. Jonathan Owen and Alexander Beecher, Department of Chemistry, Columbia University
- semiconducting quantum dots are promising for emerging technologies (tissue imaging, solid state lightning)
- Owen group found new synthesis route for making large-quantity of monodispersed CdSe quantum dots (QD)
- QDs can be prepared at 3 sizes from organic precursors
- in situ UV absorbance spectroscopy shows formation of discrete-size particles. Isolated large amounts of uniform particles with respective absorption peaks at 350, 380 and 408 nm



J. Am. Chem. Soc., 2014, 136 (30), 10645–10653

Structure of CdSe quantum dots

- single crystal diffraction from imperfect crystal of the smallest CdSe (350nm) suggests tetrahedral cutout from CdSe zinc-blende phase
- other QD sizes predicted as similar tetrahedral cutouts from CdSe terminated by {111} Cd planes
- clusters + ligands chemical formulas completed by elemental analysis, NMR and infrared absorption



CdSe$_{(350\ nm)}$ — $Cd_{31}Se_{20}(O_2CPh)_{22}(H_2NR')_{22}$

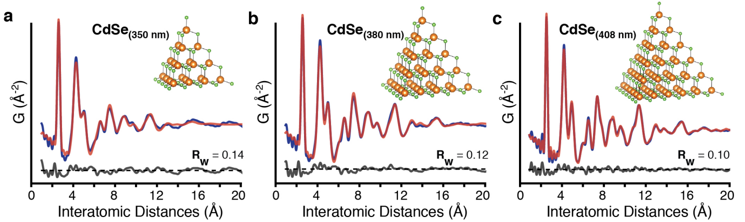
CdSe$_{(380\ nm)}$ — $Cd_{52}Se_{35}(O_2CPh)_{34}(H_2NR')_{34}$

CdSe$_{(408\ nm)}$ — $Cd_{80}Se_{56}(O_2CPh)_{48}(H_2NR')_{l}$

J. Am. Chem. Soc., 2014, 136 (30), 10645–10653

Structure of CdSe quantum dots

- experimental PDFs measured at 100K at the beamline X17A, NSLS, BNL
- tetrahedral QD models gave excellent fit with the PDFs at R_w = 0.14, 0.12, 0.10



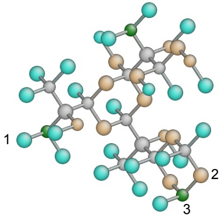
Is such PDF agreement unequivocal for tetrahedrally shaped clusters?

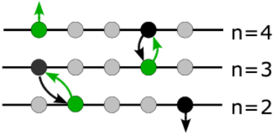
- If yes, the tetrahedral shape can be solved from the PDF data.
- small particle size, sharp PDF peaks at zinc-blende separations → PDF structure determination should be feasible.

BROOKHAVEN NATIONAL LABORATORY

Shape determination of CdSe QD-s

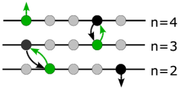
- assume structure is a cutout from CdSe zinc-blende of up to 100 atoms
- use generalized Liga algorithm to optimize cut-out size and shape



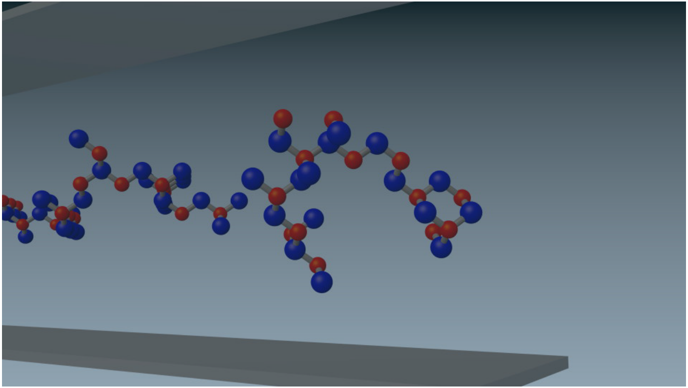


- evaluate structure cost as the PDF fit residuum R_w
  - refine lattice expansion, scaling, $U_{iso,Cd}$, $U_{iso,Se}$, δ_2
- track large number of clusters at sizes from 1 to 100 atoms
- good-cost clusters add 1 atom and are promoted to higher level
  - atoms are added as neighbors of sites with CN < 4.
- poor-cost clusters remove 1 atom and descend to lower level
  - only surface atoms (CN < 4) can be removed.
    neck atoms are protected (avoid splitting).

Shape determination of CdSe QD-s



Sample possible zinc-blende cutouts and look for structure with minimum cost (best PDF fit).

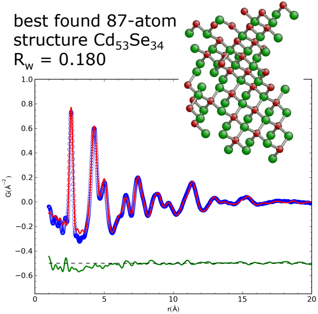


Animation by Dr. Kevin Knox

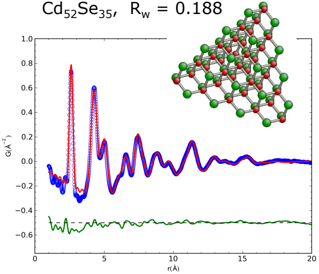
Shape determination of CdSe 380 nm

- visited ~2.5×10^5 unique clusters
- ~50% had better PDF fit than the tetrahedral model $Cd_{52}Se_{35}$

best found 87-atom structure $Cd_{53}Se_{34}$
R_w = 0.180



tetrahedral model
$Cd_{52}Se_{35}$, R_w = 0.188

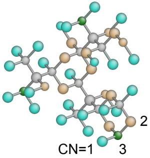


- PDF has insufficient sensitivity to particle shape (which only shows in PDF amplitudes decay)

PDF agreement is insufficient to confirm tetrahedral model → more inputs are necessary for unique structure identification.

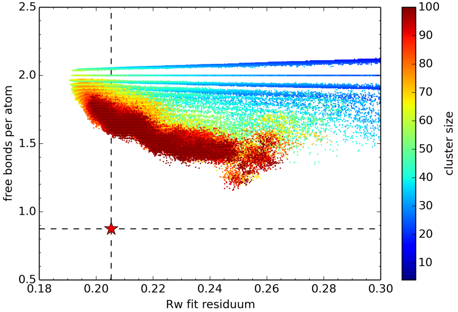
BROOKHAVEN NATIONAL LABORATORY

Overview of PDF-optimized shapes



"surface area" assessed as number of unoccupied bonds-per-atom (BPA)

$$BPA = 4 - \langle CN \rangle$$



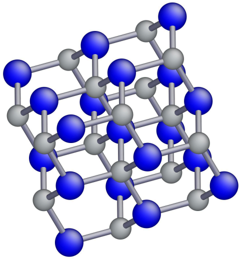
- each dot represents unique CdSe cluster visited in PDF optimizations
- dots are colored according to cluster size
- red star is the tetrahedral model $Cd_{52}Se_{35}$

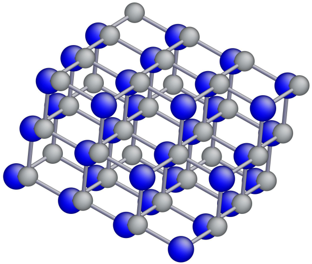
CdSe shapes optimized to PDF have much higher BPA than the tetrahedral model.

BROOKHAVEN NATIONAL LABORATORY

Shape optimization from BPA (surface area)

- cost function set to *BPA*, optimized clusters of sizes of up to 100 atoms
- minimum values of BPA found for 35 and 68 atoms

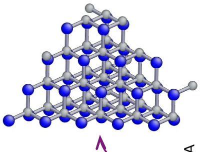


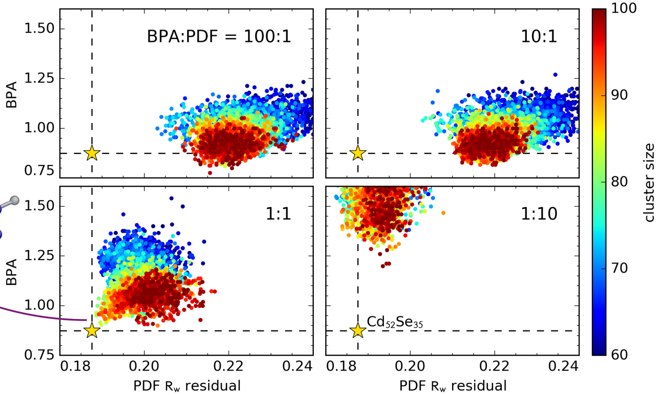


- *BPA* minimization prefers {111} facets, where surface atoms have 3 bonds.

BROOKHAVEN
NATIONAL LABORATORY

PDF + BPA optimization

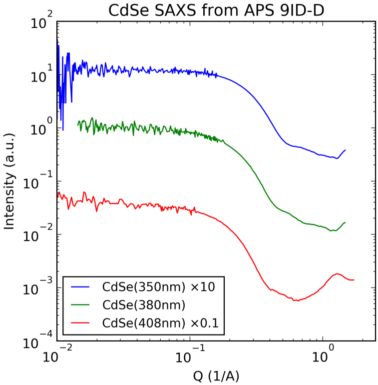




- shape optimization conducted for several BPA:PDF weight ratios
- best structures at 1:1 weight ratio are close to tetrahedra, however
- BPA is approximate and has no direct sensitivity to sample shape

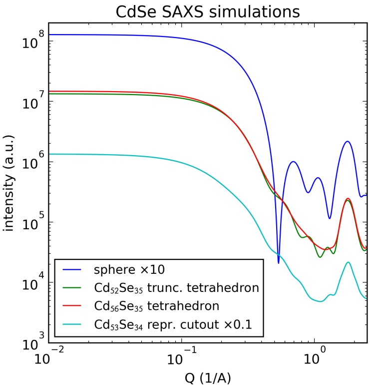
BROOKHAVEN NATIONAL LABORATORY

CdSe SAXS measurement



- SAXS data were collected at the APS 9ID-D beamline (thanks to Jan Ilavsky)
- measured 3 samples of CdSe QD-s in dilute toluene solutions
- desmeared USAXS and SAXS signal combined merged in IRENA; Q-range of [0.01, 1.2] / A

CdSe SAXS simulations

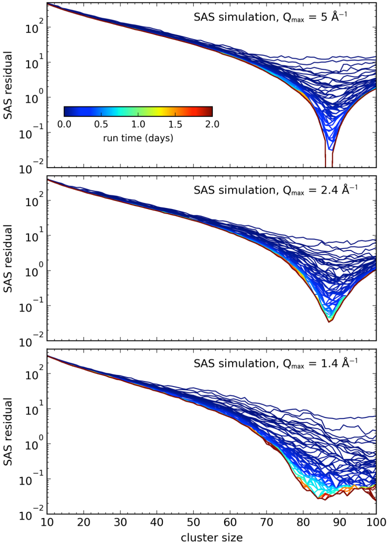


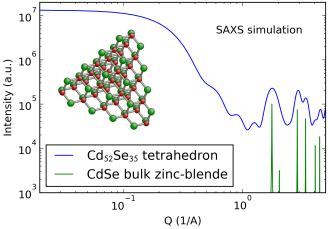
- SAXS signal has been modeled with Debye Scattering Equation

$$I(Q) = \sum_{i,j} f_i(Q) f_j(Q) \frac{\sin Q r_{ij}}{Q r_{ij}}$$

- measured Q-range is well sensitive to the shape of CdSe models
- SAXS shows clear difference between corner-truncated and full-sized CdSe tetrahedral models

CdSe shape determination from ideal SAXS





- Liga cost function set to SAXS fit residual

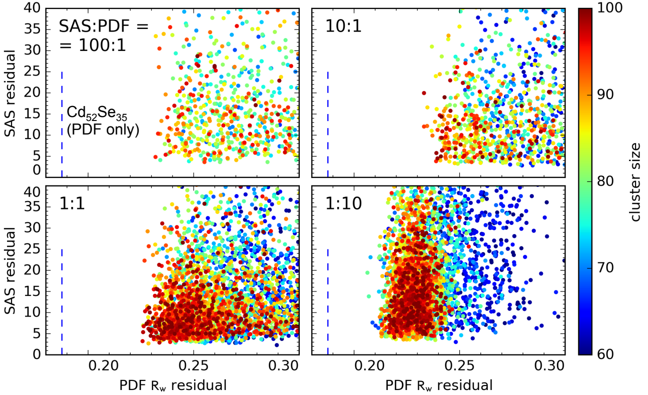
  $\log(I_{obs}) - \log(I_{calc})$ where
  $I_{calc}(Q) = A\, I_{DSE}(Q) + B$

- CdSe shape determination run from ideal SAXS data at varying Q_{max}
  - successful shape determination for Q_{max} = 5/A
  - apparent cost vs. size minimum for Q_{max} = 2.4/A
  - no convergence for Q_{max} = 1.4/A

- experimental SAXS data are not sufficient for unique shape determination.

BROOKHAVEN NATIONAL LABORATORY

Shape determination from PDF + SAS combination



- CdSe shape optimized w/r to a combined cost function: $C = W_{PDF}\, C_{PDF} + W_{SAS}\, C_{SAS}$
- C_{PDF}, C_{SAS} costs are coupled, because they both depend on the lattice expansion
- C_{PDF}, C_{SAS} give competing lattice expansion (-1% vs -0.5%)

No convincing structure found consistent with both PDF and SAS.

BROOKHAVEN
NATIONAL LABORATORY

Summary

- limited resolution in the measurements → it is crucial to combine multiple experimental and/or theoretical inputs for nanostructure solution and validation.
- excellent PDF agreement is not sufficient for a unique structure solution
- complex modeling provides verification of combined probes, detects systematic errors for shared variables.
- DiffPy-CMI – software framework for complex modeling, structure representations and manipulations, calculators for PDF, BVS, SAS, multi-component fit management, extensible, open source.
  http://www.diffpy.org

Acknowledgments

- co-developers, colleagues, users: Kevin Knox, Michael McKerns, Christopher Farrow, Dmitriy Bryndin, Jiwu Liu, Yingrui Shang, Peng Tian, Wenduo Zhou, Milinda Abeykoon, Emil Bozin, Timur Dykhne, Simon J.L. Billinge.
- DANSE, open-source crystallography software: Paul Kienzle, Paul Butler, Michael Aivazis, Vincent Favre-Nicolin.
- support by BNL LDRD 12-007.

BROOKHAVEN
NATIONAL LABORATORY