DebyeCalculator

A **GPU-accelerated open-source** python package for calculating powder diffraction, small-angle- and total scattering with the Debye scattering equation.

Frederik Lizak Johansen[1,2], Andy Sode Anker[3,4],
Ulrik Friis-Jensen[1,2], Erik B. Dam[2], Kirsten M.Ø. Jensen[1]
and Raghavendra Selvan[2]

1 Department of Chemistry & Nano-Science Center, University of Copenhagen
2 Department of Computer Science, University of Copenhagen
3 Department of Energy Conversation and Storage, Technical University of Denmark,
4 Department of Chemistry, University of Oxford, United Kingdom

Lachlan's Software Fayre @ epdic18

Padova, Italy
30 August - 2 September 2024

The Basilica of St. Anthony, Padua, Italy

Agenda for the talk

1) Brief reminder of what the Debye Scattering Equation is.
2) The motivation behind developing ***DebyeCalculator*** and why we think it's a valuable tool.
3) Explanation of what exactly "acceleration" means in this context.
4) A look at the features and functionalities currently offered by ***DebyeCalculator***.
5) Examples of how ***DebyeCalculator*** has been used in research settings.
6) Is ***DebyeCalculator*** just a tool for analysis?

7) DebyeCalculator in Action -- Demo with Examples!

<15 min

epdic18
Padova, Italy
30 August - 2 September 2024

Brief overview of the Debye Scattering Equation

- → **Context:** The equation is essential in fields where the precise understanding of atomic arrangements is crucial, particularly when traditional crystallography methods fall short due to lack of long-range order.
- → **The equation** is used to calculate the intensity of scattered X-rays or neutrons from a material by taking into account the positions of all atoms in the material.

$$I_C(Q) = \sum_{\nu\mu} f_\nu(Q) f_\mu(Q) \frac{\sin(Q r_{\nu\mu})}{Q r_{\nu\mu}}$$

$$S(Q) = \frac{I_C(Q) - \left[\langle f(Q)^2 \rangle - \langle f(Q) \rangle^2\right]}{\langle f(Q) \rangle^2}.$$

$$F(Q) = Q(S(Q) - 1)$$

$$G(r) = \left(\frac{2}{\pi}\right) \int_{Q_{\min}}^{Q_{\max}} F(Q) \sin(Qr)\, \mathrm{d}Q$$

epdic18
Padova, Italy
30 August - 2 September 2024

Motivation behind *DebyeCalculator*

Thanks to Scardi and Wright for brilliantly motivating this in the opening of EPDIC18!

Auto-differentiable calculations

→ The use of the Debye Scattering Equation as "cost"-function in Machine Learning.

In the end, our motivation became: to democratize high-speed scattering calculations, enabling rapid, accessible, and scalable analysis of materials.

epdic18
Padova, Italy
30 August - 2 September 2024

What do we mean when we say “accelerated”?

→ We mean focusing the following key aspects:

1. **Pre-computations (FFP & TRI-U)**
2. **Vectorization of key operations (PyTorch)**
3. **Parallelism on GPU (CUDA)**
4. **Pseudo batching (Slicing and splitting)**
5. Selective computation

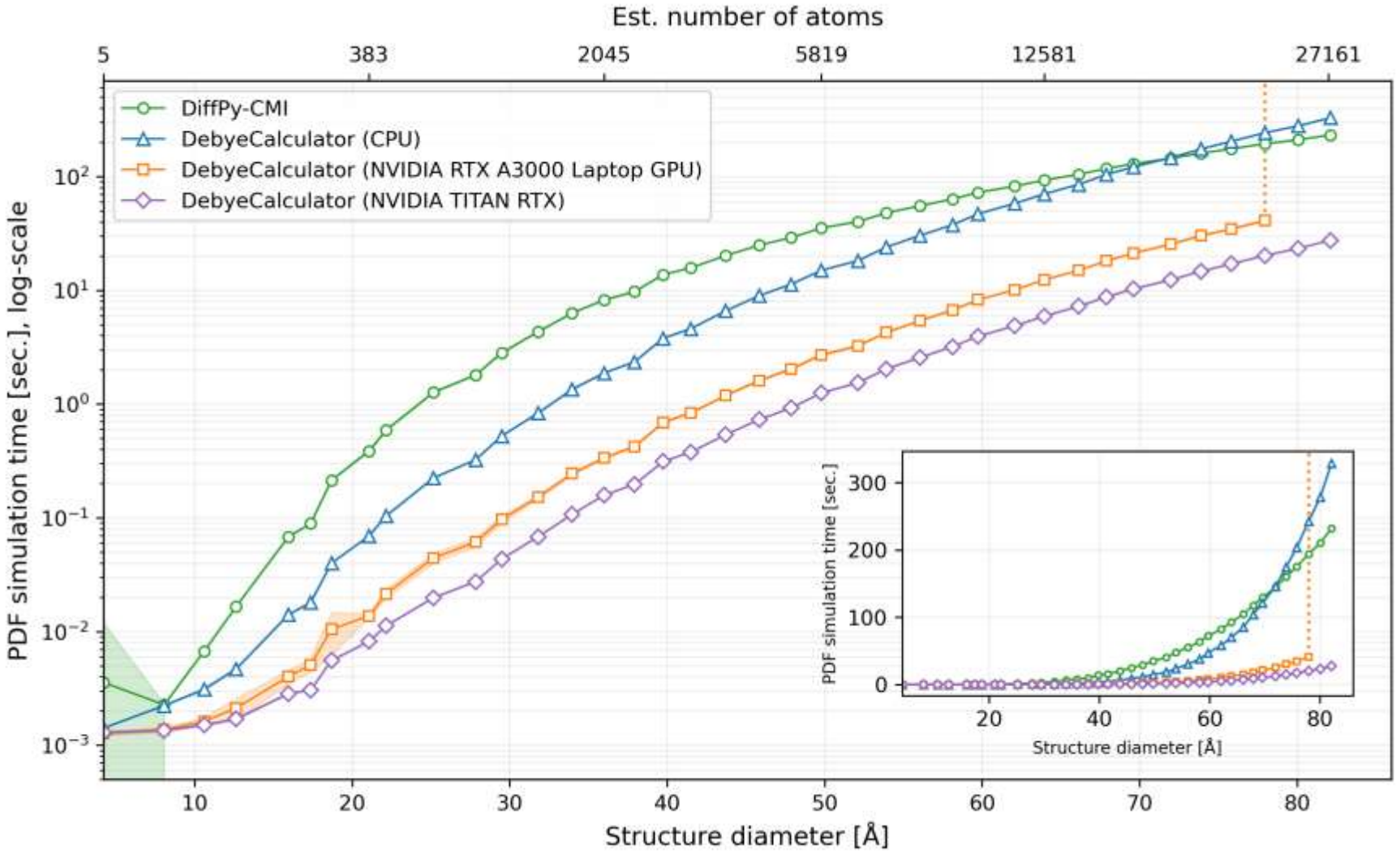
(FUTURE) Parallelism on cluster of nodes?

$$I(Q) = \sum_i \sum_j \frac{f_i f_j \sin(Q r_{ij})}{Q r_{ij}}$$

```
dists = torch.norm(...)[*triu_indices].split(self.batch_size)
iq = torch.zeros(...).to(device=self.device)
sinc = torch.sinc(d[mask] * self.q / torch.pi)
```

epdic18
Padova, Italy
30 August - 2 September 2024

What do we mean when we say “accelerated”?



NVIDIA CUDA

- Pseudo batch size: 10,000
- 10 simulations per sample.

NVIDIA RTX A3000 Laptop GPU, 6GB RAM: ~5x speed-up compared to CPU.

NVIDIA TITAN RTX, 24GB RAM : ~10x speed-up compared to CPU.

epdic18 Padova, Italy 30 August - 2 September 2024

Journal of Open Source Software

[REVIEW]: A GPU-Accelerated Open-Source Python Package for Calculating Powder Diffraction, Small-Angle-, and Total Scattering with the Debye Scattering Equation #6024

New issue

Closed editorialbot opened this issue on Nov 6, 2023 · 74 comments

editorialbot commented on Nov 6, 2023 • edited · Collaborator

Submitting author: @AndySAnker (Andy Sode Anker)
Repository: https://github.com/FrederikLizakJohansen/DebyeCalculator
Branch with paper.md (empty if default branch):
Version: v1.0.7
Editor: @phibeck
Re
Au

elena-pascal commented on Nov 7, 2023 • edited

Review checklist for @elena-pascal

Conflict of interest

- [ ] I confirm that I have read the JOSS conflict of interest (COI) policy and that: I have no COIs with perceived COIs have been waived by JOSS for the purpose of this review.

KedoKudo commented on Nov 28, 2023

I created a few issues after the first round of review:

- [JOSS review] benchmark on Apple silicon FrederikLizakJohansen/DebyeCalculator#16
- [JOSS review] GoogleColab example does not show plots FrederikLizakJohansen/DebyeCalculator#15
- [JOSS review] Add documentation and publish it FrederikLizakJohansen/DebyeCalculator#14
- [JOSS review] Add tags and release on Github FrederikLizakJohansen/DebyeCalculator#13

I will continue the review after these issues are addressed

About

A vectorised implementation of the Debye Scattering Equation on CPU and GPU

- Readme
- Apache-2.0 license
- Cite this repository
- Activity
- 18 stars
- 2 watching
- 4 forks

JOSS
The Journal of Open Source Software

A GPU-Accelerated Open-Source Python Package for Calculating Powder Diffraction, Small-Angle-, and Total Scattering with the Debye Scattering Equation

Frederik L. Johansen, Andy S. Anker, Ulrik Friis-Jensen, Erik B. Dam, Kirsten M. Ø. Jensen, and Raghavendra Selvan

1 Department of Chemistry & Nano-Science Center, University of Copenhagen, Denmark 2 Department of Computer Science, University of Copenhagen, Denmark 3 Department of Energy Conversion and Storage, Technical University of Denmark, Denmark 4 Department of Chemistry, University of Oxford, United Kingdom ¶ Corresponding author * These authors contributed equally.

DOI:

Software:
- Review
- Repository
- Archive

Editor:

Reviewers:

Submitted:
Published:

License:

Summary

The Debye scattering equation, derived in 1915 by Peter Debye, is used to calculate scattering intensities from atomic structures considering the position of each atom in the structure

$$I(Q) = \sum_{n=1}^{N}\sum_{m=1}^{N} b_n b_m \frac{\sin(Qr_{nm})}{Qr_{nm}} \quad (1)$$

In this equation Q is the momentum transfer of the scattered radiation, N is the number of atoms in the structure, and r_{nm} is the distance between atoms n and m. For X-ray radiation, the atomic form factor, b, depends strongly on Q and is usually denoted as $f(Q)$, but for neutrons, b is independent of Q and referred to as the scattering length. The Debye scattering equation can be used to compute the scattering pattern of any atomic structure and is commonly used to study both crystalline and non-crystalline materials with a range of scattering techniques like powder diffraction (PD), total scattering (TS) with pair distribution function (PDF) analysis, and small-angle scattering (SAS). Although the Debye scattering equation is extremely versatile, the computation of the double sum, which scales $\mathcal{O}(N^2)$, has limited the practical use of the equation.

With the advancement in computer hardware, analysis of larger structures is now feasible using the Debye scattering equation. Modern central processing units (CPUs), ranging from tens to hundreds of cores offer an opportunity to parallelise computations, significantly enhancing compute efficiency. The same goes for graphics processing units (GPUs), which are designed with multiple cores acting as individual accelerated processing units that can work on different tasks simultaneously. In contrast, CPUs usually have fewer cores optimised for more general-purpose computing. This means that a GPU can execute multiple simple instructions in parallel, while a CPU might handle fewer parallel tasks. Therefore, GPUs are better suited for calculations such as the Debye scattering equation, where many computations can be performed simultaneously. Taking advantage of such GPU acceleration could yield computational speeds that surpass those of even the most advanced multi-core CPUs by orders of magnitude. We introduce a GPU-accelerated open-source Python package, named DebyeCalculator, for rapid calculation of the Debye scattering equation from chemical structures represented as .xyz or .cif files. The xyz-format is commonly used in materials chemistry for the description of discrete particles and simply

epdic18
Padova, Italy
30 August - 2 September 2024

Installation is easy

Prerequisites

`DebyeCalculator` requires Python version >=3.7, <3.12. If needed, create an environment with any of these Python versions:

```
conda create -n debyecalculator_env python=3.9
```

```
conda activate debyecalculator_env
```

Before installing the `DebyeCalculator` package, ensure that you have PyTorch installed. Follow the instructions on the official PyTorch website to install the appropriate version for your system: PyTorch Installation Guide.

NOTE: Installing an earlier version of PyTorch (<=1.13.1) will be necessary if you're running Python 3.7, since the latest PyTorch version requires Python 3.8 or higher.

epdic18
Padova, Italy
30 August - 2 September 2024

Installation is easy

PyTorch

PyTorch Build	**Stable (2.4.0)**		Preview (Nightly)	
Your OS	**Linux**	Mac	Windows	
Package	Conda	**Pip**	LibTorch	Source
Language	**Python**		C++ / Java	
Compute Platform	CUDA 11.8	CUDA 12.1	**CUDA 12.4**	ROCm 6.1	CPU
Run this Command:	`pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu124`			

Installation is easy

Install with pip

Run the following command to install the **DebyeCalculator** package. (**Requires**: Python >=3.7, <3.12)

```
pip install debyecalculator
```

Install locally

Clone the repo

```
git clone https://github.com/FrederikLizakJohansen/DebyeCalculator.git
```

Run the following command to install the **DebyeCalculator** package. (**Requires**: Python >=3.7, <3.12)

```
python -m pip install .
```

epdic18
Padova, Italy
30 August - 2 September 2024

Features and functionalities

To start using *DebyeCalculator*, all it takes is a single **import statement**:

```python
from debyecalculator import DebyeCalculator

# Create an instance 'calc' of DebyeCalculator
calc = DebyeCalculator(
    qmin=0.0,         # Minimum Q value (1/Å).
    qmax=30.0,        # Maximum Q value (1/Å).
    qstep=0.1,        # Step size for Q (1/Å), Nyquist-sampled by default.
    qdamp=0.0,        # Dampening factor.

    rmin=0.0,         # Minimum R value (Å).
    rmax=20.0,        # Maximum R value (Å).
    rstep=0.01,       # Step size for R (Å).
    rthres=0.0,       # Lower threshold for radial distance (Å) in G(r) calculations.

    biso=0.0,         # Isotropic atomic displacement factor (Debye-Waller factor).
    device='cuda',    # Compute on 'cuda' for GPU or 'cpu' for CPU.
    batch_size=None, # Batch size for pairwise distance calculations (auto-determined if not set).
    lorch_mod=False, # Apply Lorch modification factor in G(r) calculations.
    rad_type='xray', # Type of radiation for scattering: 'xray'/'x' or 'neutron'/'n'.
    profile=False,    # Enable profiling for performance analysis.
)
```

- Set class parameters based on given input or defaults
- Setup scattering parameters (e.g., Q-values, r-values) and hardware parameters
- Load atomic form factor coefficients
- Setup form factor calculation based on radiation type

epdic18
Padova, Italy
30 August - 2 September 2024

Features and functionalities: **Basics**

Sequential calculations of scattering for one or more structures at a time.

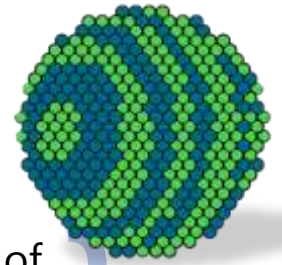
```
DebyeCalculator instance:
Q-min:          1.00
Q-max:         30.00
Q-step:         0.16
Q-damp:         0.04

r-min:          0.00
r-max:         20.00
r-step:         0.01
r-thres:        0.00

B-iso:          0.30

rad_type:      xray
lorch_mod:     False

batch_size:    10000
device:        cuda
profile:       False
```

```
q, iq = calc.iq(...)
q, sq = calc.sq(...)
q, fq = calc.fq(...)
r, gr = calc.gr(...)
```

.xyz

structure tuples

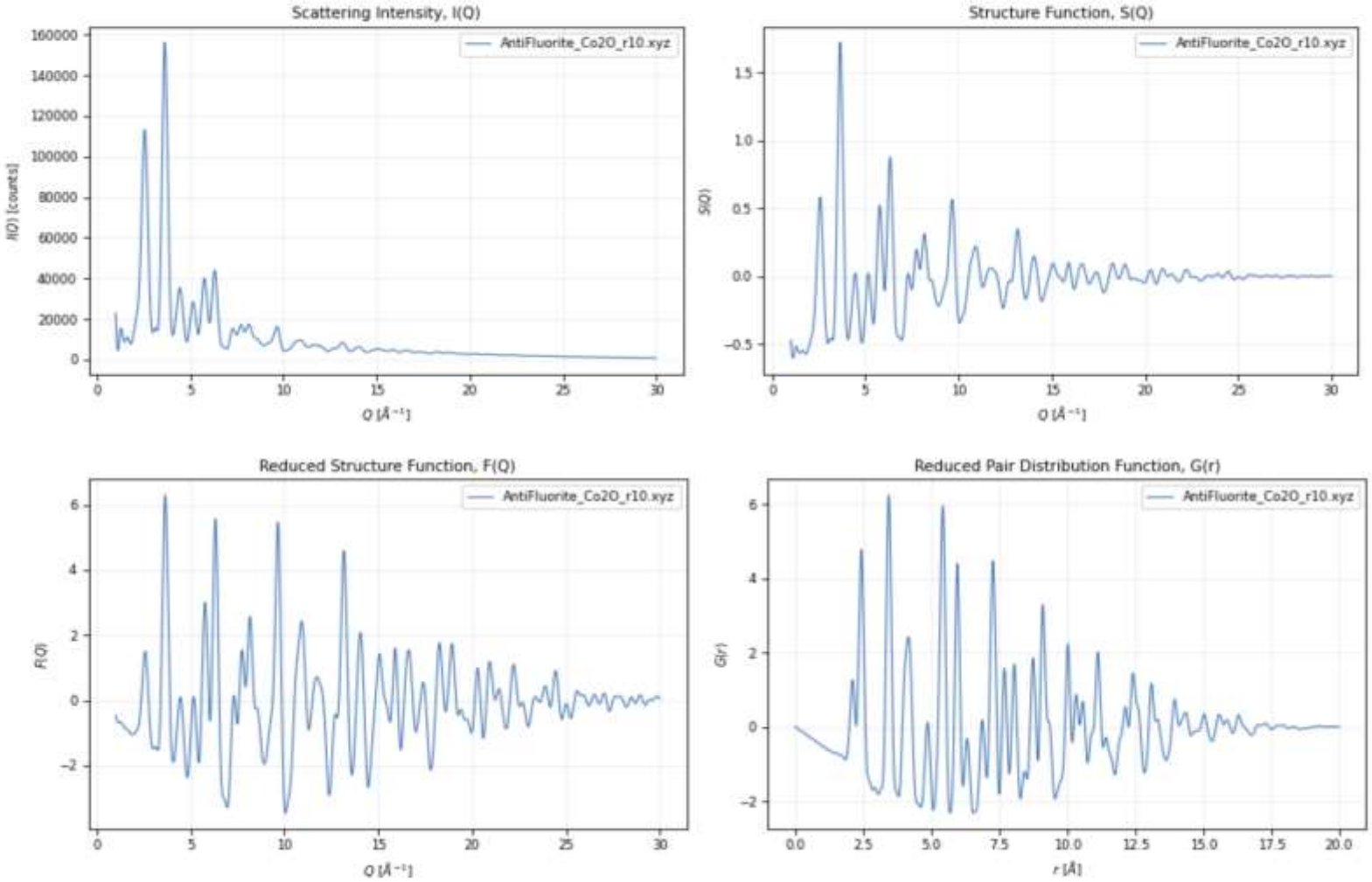
ASE Atoms



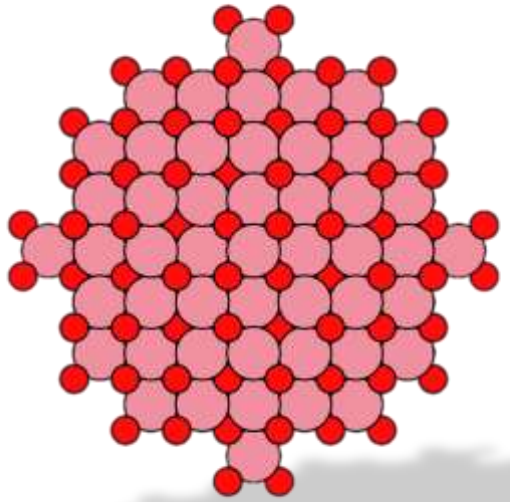
.cif → Implicit generation of spherical particle structures from list of radii [Å]

epdic18
Padova, Italy
30 August - 2 September 2024

Features and functionalities: **Basics**



AntiFluorite Co_2O



epdic18
Padova, Italy
30 August - 2 September 2024

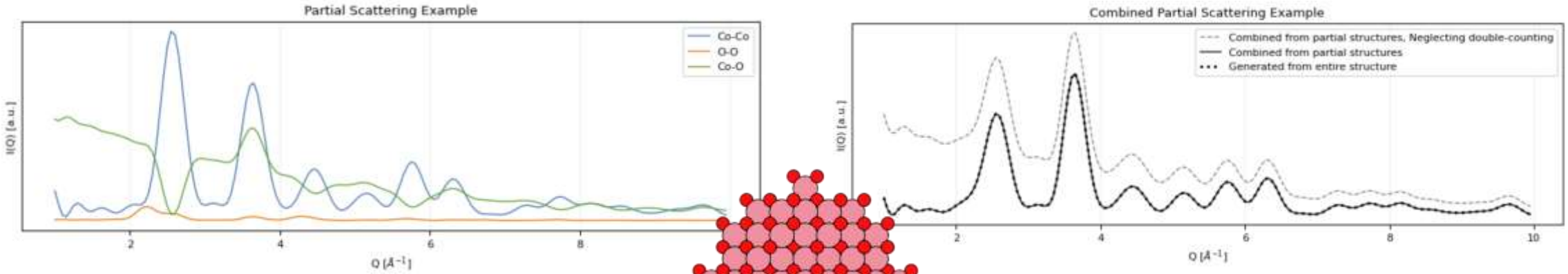
Features and functionalities: **Partial Scattering**

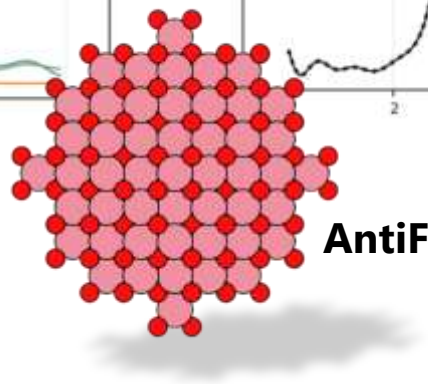
`DebyeCalculator` allows user to extract the partial scattering of specific pairs of atomic species within structures.

```
q, iq_XX = calc.iq("path/to/nanoparticle.xyz", partial = "X-X")
q, iq_YY = calc.iq("path/to/nanoparticle.xyz", partial = "Y-Y")
q, iq_XY = calc.iq("path/to/nanoparticle.xyz", partial = "X-Y", include_self_scattering = False)
```

$$I(Q) = \sum_i \sum_j \frac{f_i f_j \sin(Q r_{ij})}{Q r_{ij}}$$

(*) **Note!** that to combine signals from partial scattering, you risk double-counting some of the interactions between atoms -- so be careful!

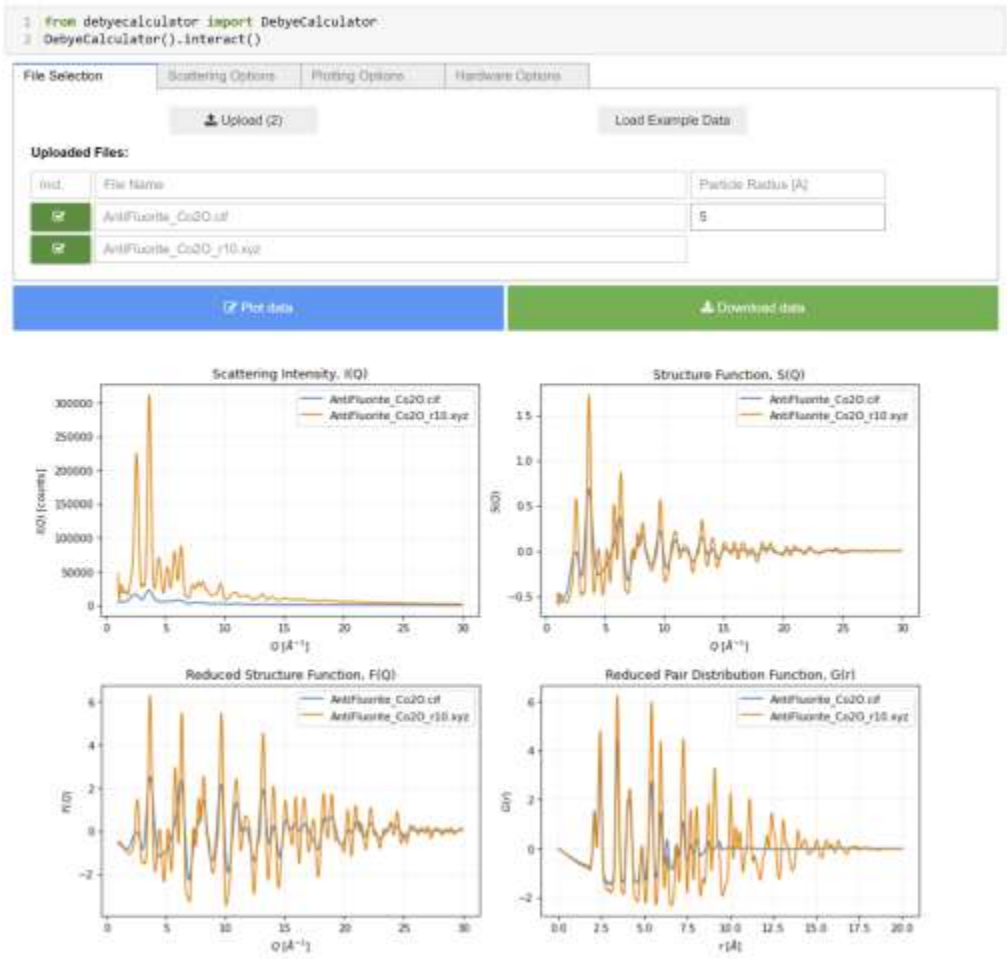


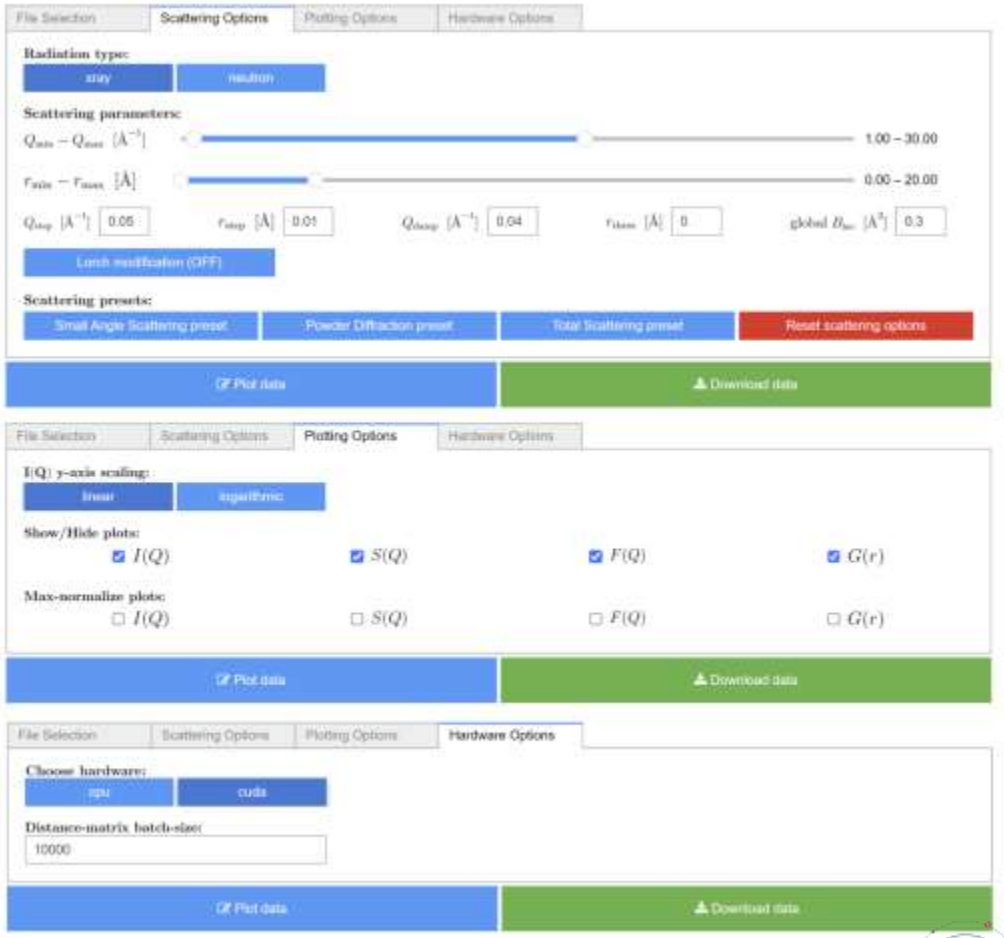


AntiFluorite Co_2O

epdic18
Padova, Italy
30 August - 2 September 2024

Features and functionalities: **Interact-Mode**





epdic18
Padova, Italy
30 August - 2 September 2024

Features and functionalities: **Developer-tool(s)**

DebyeCalculator comes shipped with a **local profiler class**, that can be activated (and changed in the code) to profile different parts of the code.

```python
calc = DebyeCalculator(profile=True, device='cuda', batch_size=10_000)

for _ in range(1_000):
    output = calc.gr(structure_path, radii = 10.0)

print(calc.profiler.summary(prefix="Timing per PDF generation:\n"))
```

As simple as:

```python
if self.profile:
    self.profiler.time('G(r)')
```

epdic18
Padova, Italy
30 August - 2 September 2024

DebyeCalculator in research:

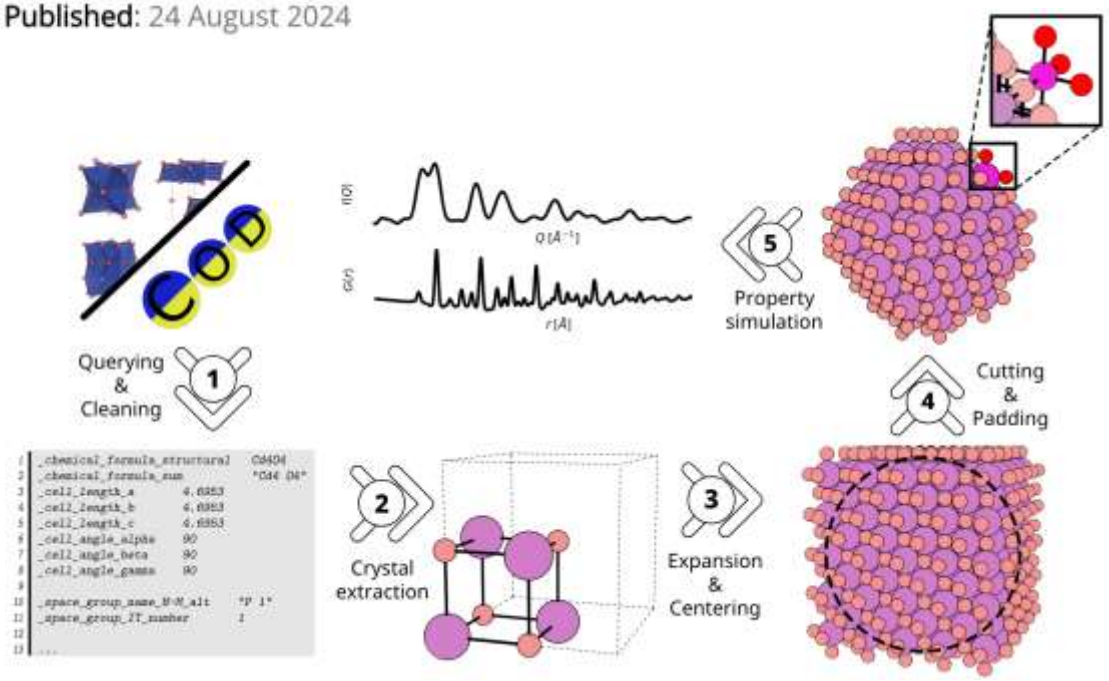
Association for Computing Machinery

KDD2024 BARCELONA, SPAIN

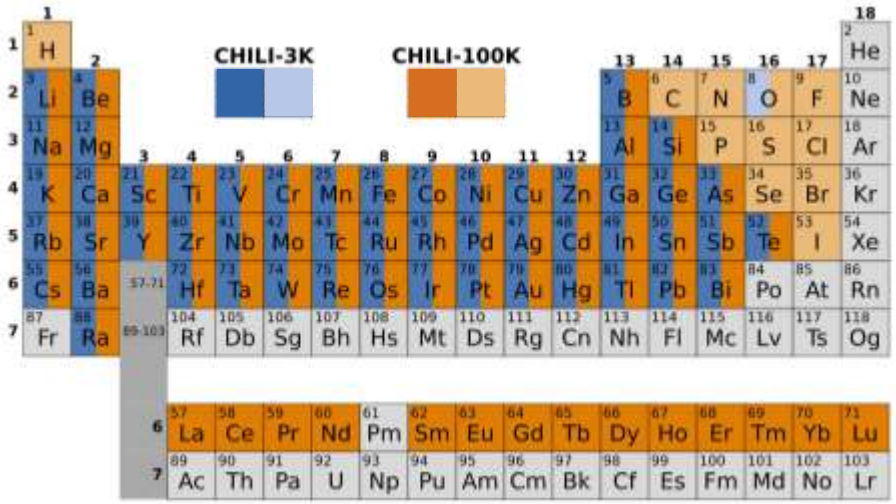
CHILI: Chemically-Informed Large-scale Inorganic Nanomaterials Dataset for Advancing Graph Machine Learning

Authors: Ulrik Friis-Jensen, Frederik L. Johansen, Andy S. Anker, Erik B. Dam, Kirsten M. Ø. Jensen, Raghavendra Selvan

Published: 24 August 2024



Dataset	# of graphs	# of nodes	# of edges	Generation time
CHILI-3K	3,180	6,959,085	49,624,440	02h51m29s
CHILI-100K	104,408	183,398,463	1,251,841,365	68h25m12s



https://doi.org/10.1145/3637528.3671538

epdic18 Padova, Italy 30 August - 2 September 2024

DebyeCalculator in research:

[Submitted on 23 Jul 2024]

Signatures of paracrystallinity in amorphous silicon

Louise A. M. Rosset, David A. Drabold, Volker L. Deringer

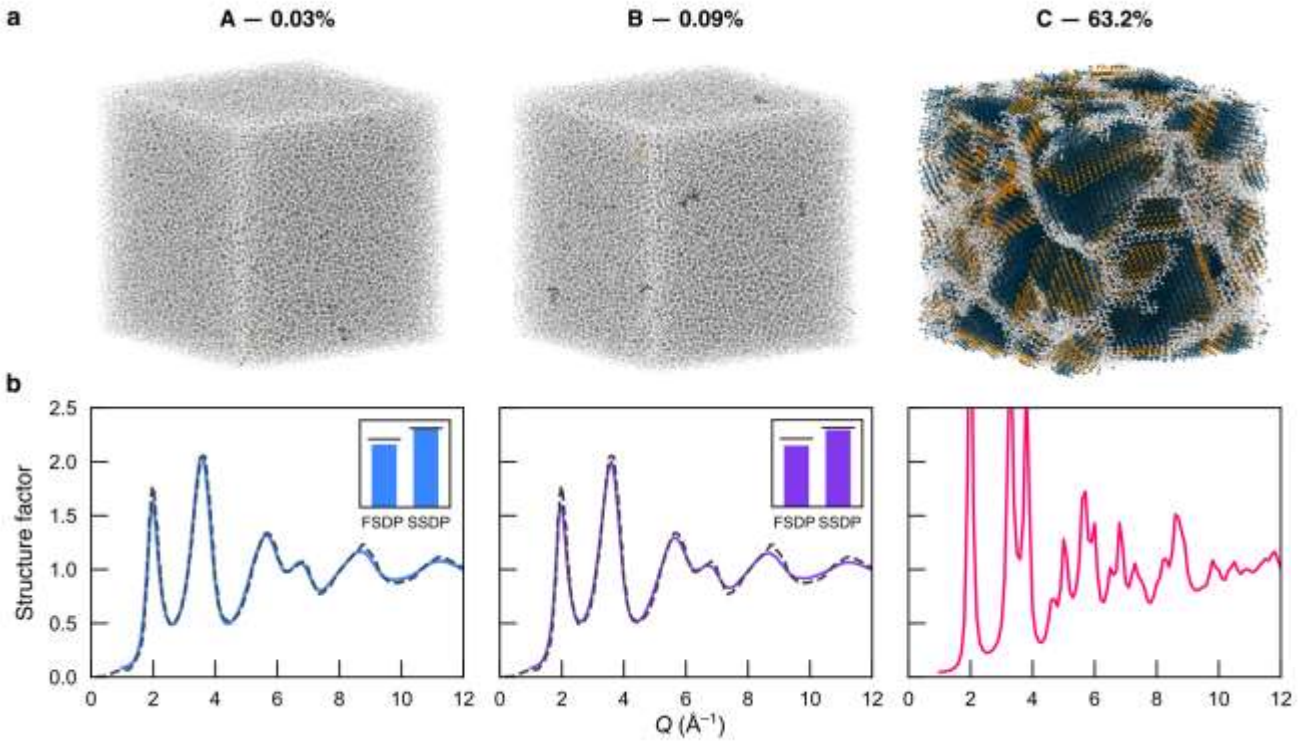


Figure 4: Three a-Si structural models of 100,000 atoms with increasing paracrystallinity. The first model, labeled **A**, is taken from Ref. 4, while the other two were generated as part of the present study by melt-quenching at 10^{11} K/s (**B**) and 10^{10} K/s (**C**), respectively. (a) Structure visualizations color-coded by PTM as in Fig. 1. (b) Computed structure factor for each structure, using the DEBYECALCULATOR package.[44] Black dashed lines indicate the experimental data from Ref. 36. Insets show the agreement of the predicted first and second sharp diffraction peaks (bars) with experimental data from Ref. 36 (black lines).

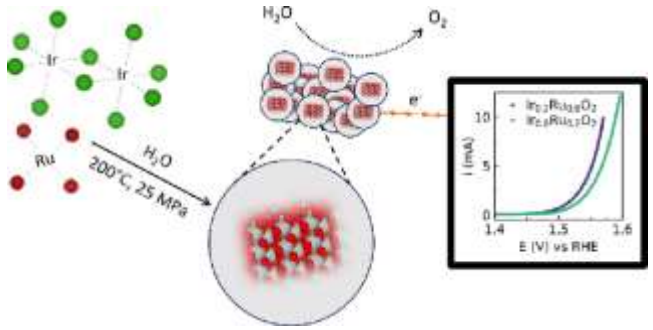
https://doi.org/10.48550/arXiv.2407.16681

DebyeCalculator in research:

ARTICLE | August 16, 2024

Formation Mechanism and Hydrothermal Synthesis of Highly Active $Ir_{1-x}Ru_xO_2$ Nanoparticles for the Oxygen Evolution Reaction

Andreas Dueholm Bertelsen, Magnus Kløve, Nils Lau Nyborg Broge, Martin Bondesgaard, Rasmus Baden Stubkjær, Ann-Christin Dippel, Qinyu Li, Richard Tilley, Mads Ry Vogel Jørgensen, and Bo Brummerstedt Iversen*



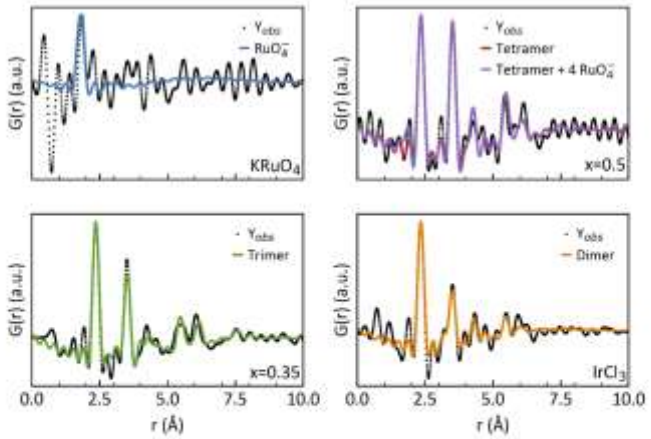


Figure S18 Experimental PDFs at room temperature of four different compositons of $KRuO_4$ and $IrCl_3$. Simulated PDFs constructed using the Debye Calculator python package[4] of structures hypothesized to be present are presented. Dimer, trimer and tetramer refers to the size of the $IrCl_4$ cluster modelled. Data on pure $IrCl_3$ and pure $KRuO_4$ are collected at P21.1 in September 2021. Data on x=0.5 and x=0.35 are collected at P21.1 in May 2022.

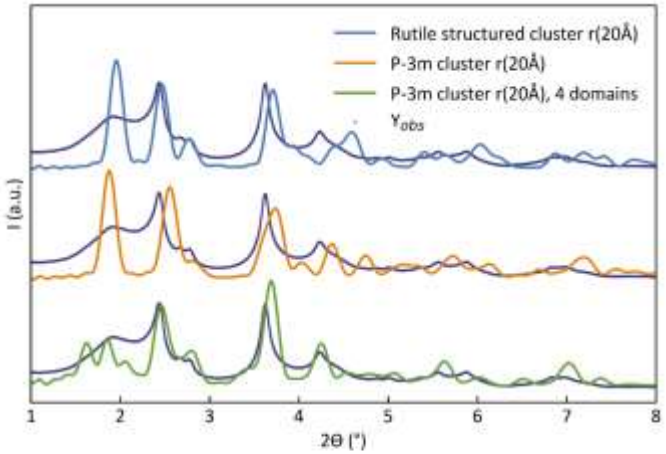


Figure S26 a) Experimental diffraction pattern of x=0.8 with simulated diffraction patterns of nanoparticle-sized cutouts of the rutile structure and the $P\bar{3}m$ structure. Scattering patterns are constructed using the Debye Calculator Python package.

https://doi.org/10.1021/jacs.4c04607

epdic18
Padova, Italy
30 August - 2 September 2024

DebyeCalculator in research:

ChemRxiv™

Order in Disorder: Tracking the Formation of High Entropy Solid Solutions and High Entropy Intermetallics by in situ PXRD/XAS

Nicolas Schlegel, Stefanie Punke, Christian M. Clausen, Ulrik Friis-Jensen, Adam F. Sapnik, Dragos Stoian, Olivia Aalling-Frederiksen, Divyansh Gautam, Jan Rossmeisl, Rebecca K. Pittkowski, Matthias Arenz, Kirsten M. Ø. Jensen

23 August 2024, Version 1

From High-entropy Alloy to High-entropy Intermetallic: An in situ XRD/XAS Study of PtFeCoNiPd Nanoparticle Formation
Nicolas Schlegel
MS02 - In situ and operando studies
01/09/2024
GROUND FLOOR - 2

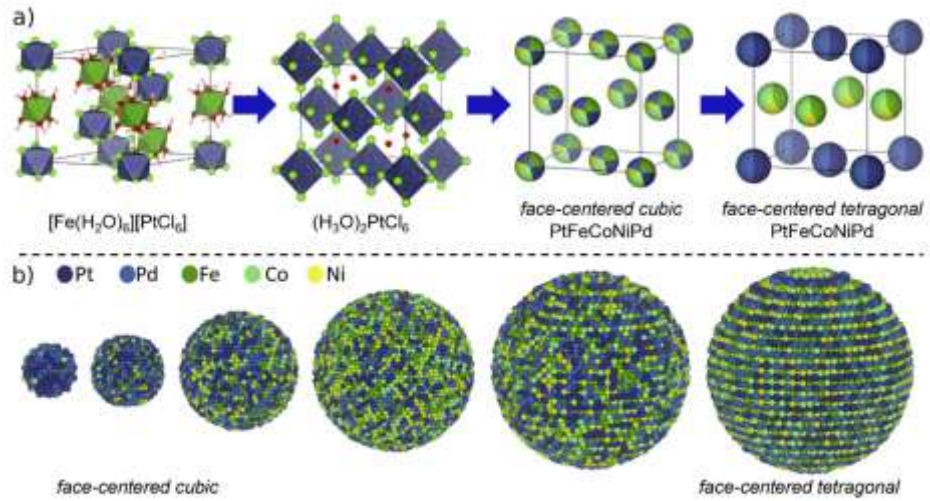


***Figure 10**: a) The transformation of the major unit cell of a PtFeCoNiPd sample during the synthesis from precursor salt to reduced intermetallic; b) Schematic illustration of particle growth and ordering throughout the synthesis of a PtFeCoNiPd sample.*

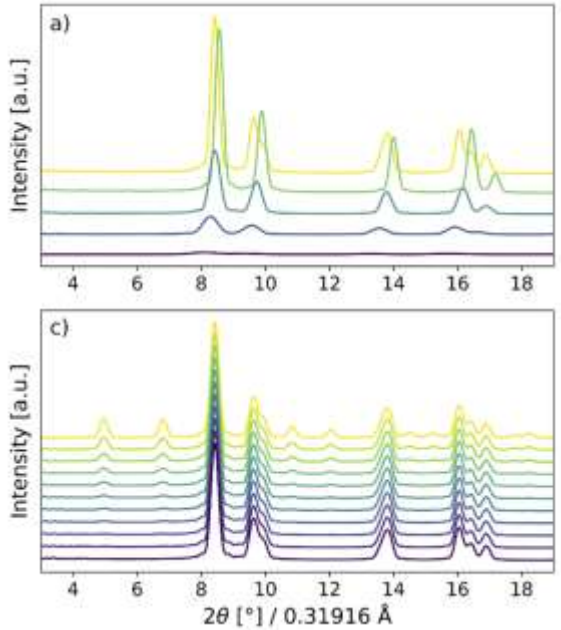
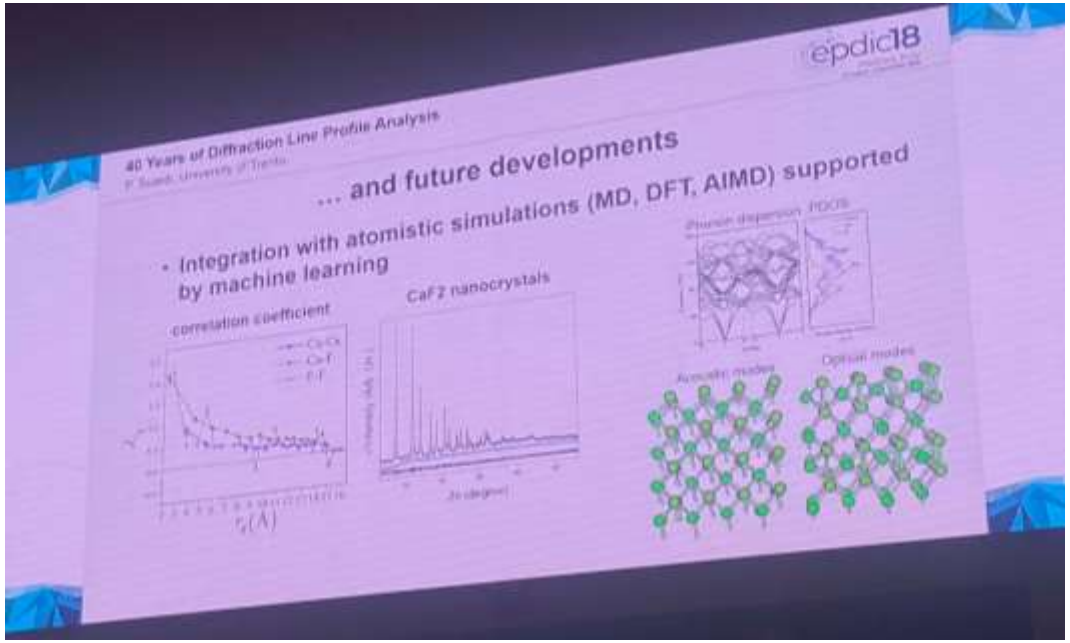


Figure S19: Calculated diffraction patterns of the simulated nanoparticles: a) during growth; b) during ordering. In a), the purple to yellow patterns correspond to the particles in row a to e in **Table S6**. In b), the purple to yellow pattern corresponds to the particles in row f to o in **Table S6.**

epdic18
Padova, Italy
30 August - 2 September 2024

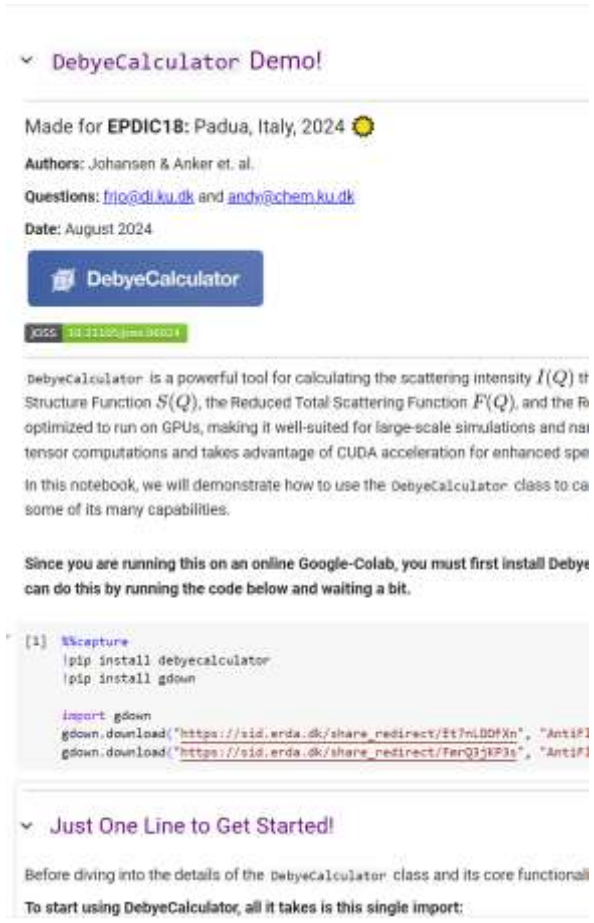
https://doi.org/10.26434/chemrxiv-2024-k1v88

How can we inspire to innovate and explore new ideas?



epdic18
Padova, Italy
30 August - 2 September 2024

Time for the demo!



DebyeCalculator Demo!

Made for **EPDIC18:** Padua, Italy, 2024

Authors: Johansen & Anker et. al.

Questions: frio@di.ku.dk and andy@chem.ku.dk

Date: August 2024

DebyeCalculator

joss

DebyeCalculator is a powerful tool for calculating the scattering intensity $I(Q)$ t
Structure Function $S(Q)$, the Reduced Total Scattering Function $F(Q)$, and the R
optimized to run on GPUs, making it well-suited for large-scale simulations and na
tensor computations and takes advantage of CUDA acceleration for enhanced spe

In this notebook, we will demonstrate how to use the DebyeCalculator class to ca
some of its many capabilities.

**Since you are running this on an online Google-Colab, you must first install Debye
can do this by running the code below and waiting a bit.**

```
[1] %%capture
    !pip install debyecalculator
    !pip install gdown

    import gdown
    gdown.download("https://sid.erda.dk/share_redirect/Et7nLDDfXn", "AntiFl
    gdown.download("https://sid.erda.dk/share_redirect/FerQ3jKP3s", "AntiFl
```

Just One Line to Get Started!

Before diving into the details of the DebyeCalculator class and its core functionali

To start using DebyeCalculator, all it takes is this single import:

tinyurl.com/
debyedemo

DebyeCalculator

Actually works on mobile if you are an Android user!

**github.com/
FrederikLizakJohansen/
DebyeCalculator**

epdic18
Padova, Italy
30 August - 2 September 2024