doi:10.1017/S1431927620023326
Microsc. Microanal. 26 (Suppl 2), 2020

A Python Based Open-source Multislice Simulation Package for Transmission Electron Microscopy

Hamish Brown, Phillipp Pelz, Colin Ophus and Jim Ciston

Lawrence Berkeley National Laboratory, Berkeley, California, United States

Strong electron multiple scattering in transmission electron microscopy (TEM) means that most quantitative work in electron microscopy requires accompanying image simulations for reliable interpretation. Recent years have seen a number of different TEM simulation packages released, many of them open source. Most are written in programming languages such as C++ [1-2] and Fortran [3] which results in the fastest possible run-time for simulations but means that modification and extension of the code is difficult for non-expert programmers. Inspired by the popular success of open-source electron microscopy packages such as HyperSpy [4] and py4DSTEM [5], which are written in the more accessible Python programming language, in this presentation we introduce an open-source purely python based TEM simulation package which can run on graphical processing units (GPUs) using the Python pytorch library [6].

The package is capable of Fig 1.(a) convergent beam electron diffraction (CBED), Fig 1. (b) scanning transmission electron microscopy (STEM) and Fig 1. (c) 4D-STEM simulations. Ionization based simulations such as STEM-EELS are implemented using the open-source Flexible Atomic Code (FAC) [7] to calculate ionization cross sections for atoms of interest. Recent algorithmic advances such as the PRISM algorithm, which economized on the number of multislice operations necessary by calculating the scattering matrix for the sample of interest, are implemented both for conventional STEM [8] and STEM-EELS [9] in the package. An example simulation of an O map of a $SrTiO_3$-CeO_2 interface structure, which took 1.5 hours in our python package but would likely have taken days in conventional multislice, is shown in Fig. 1(d). Being a python based package TEM image simulations can be run from an IPython notebook and easily leverage existing Python packages, such as the Atomic Simulation Environment (ASE) [10] and the Pymatgen (Python Materials Genomics) library [11], to build input objects for TEM simulation, or py4DSTEM for analysis of 4D-STEM data [5].

This presentation will briefly overview some of the underlying simulation algorithms and demonstrate the example uses of the code [12].

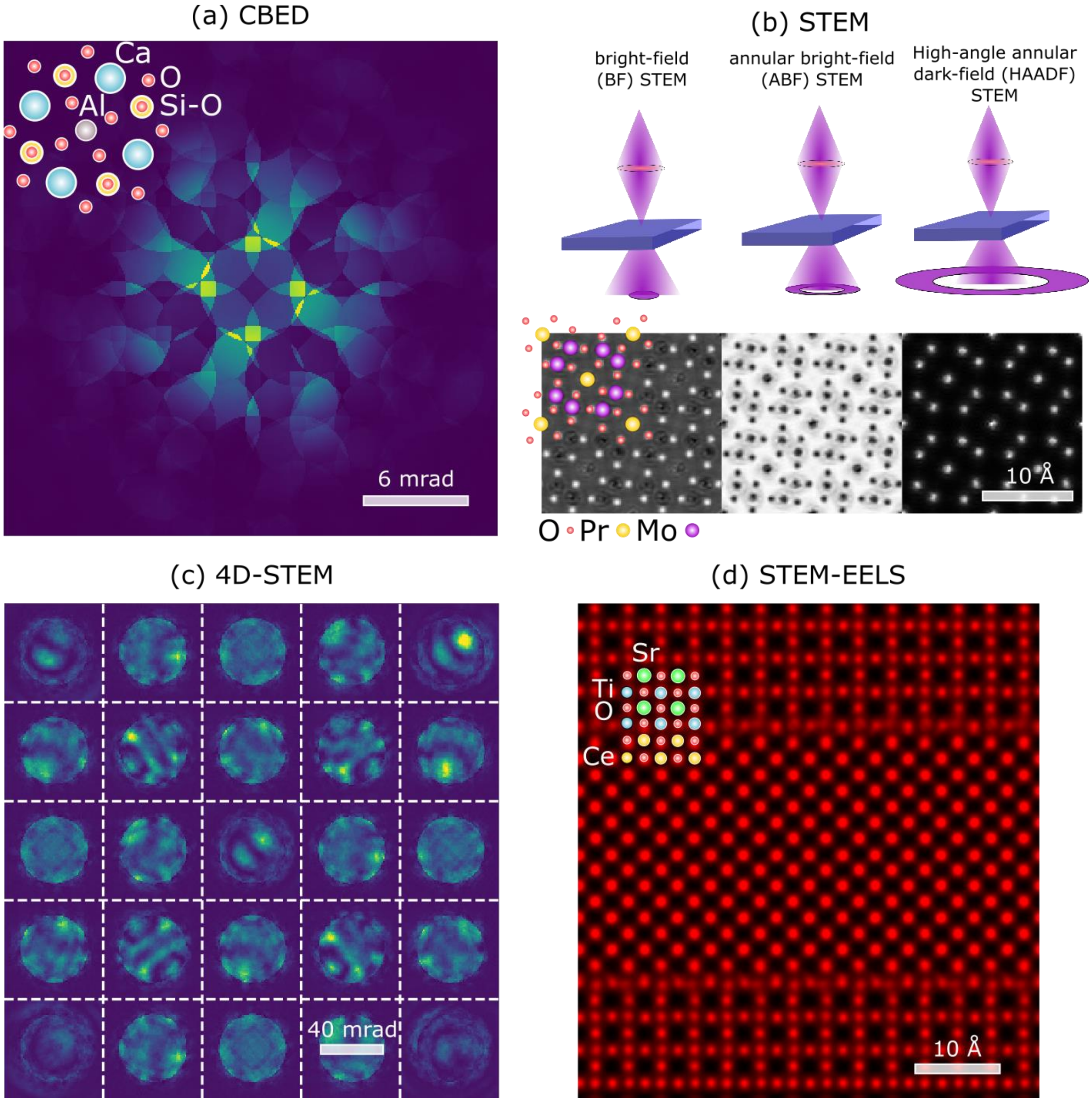

Figure 1. Multislice simulations of (a) a convergent beam electron diffraction (CBED) pattern for an $Al_2Ca_2O_7Si$ crystal (b) STEM simulations for different detector configurations for a $PrMo_6O_{12}$ structure and (c) a montage of 4D-STEM diffraction patterns for the structure in (b) and (d) a STEM-EELS simulation for elemental mapping of O using the K-edge for a $SrTiO_3$-CeO_2 interface

References

[1] Lobato, I, et al. Ultramicroscopy 156 (2015) 9.
[2] Pryor, A, et al. Adv Struct Chem Imaging **3** (2017) 15.
[3] Allen, L. J., et al. Ultramicroscopy **151** (2015) 11.
[4] de la Peña, F., et al. Microscopy and Microanalysis **23.S1** (2017) 214.
[5] Savitzky, B. H. et al. Microscopy and Microanalysis **25** (2019) 124.
[6] Paszke, A. et al. arXiv:1912.01703 (2019).
[7] https://github.com/flexible-atomic-code/fac
[8] Ophus, C. Adv Struct Chem Imaging **3** (2017) 13.

[9] Brown, H. G., et al. PRR **1** (2019) 033186.
[10] Larsen, A. H., et al. J. Condens. Matter Phys. **29.27** (2017) 273002.
[11] Ong, S. P., et al. Comput. Mat. Sci. **68** (2013) 314.
[12] Work at the Molecular Foundry was supported by the Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. CO acknowledges additional support from the U.S. Department of Energy Early Career Research

https://doi.org/10.1017/S1431927620023326 Published online by Cambridge University Press