PyNX: high-performance coherent imaging toolkit for CDI, Ptychography and Holo-Tomography

V. Favre-Nicolin[1,2,@]

[1] ESRF- The European Synchrotron, Grenoble,France [2] Univ. Grenoble Alpes, INAC-SP2M, Grenoble, France

@ favre@esrf.eu

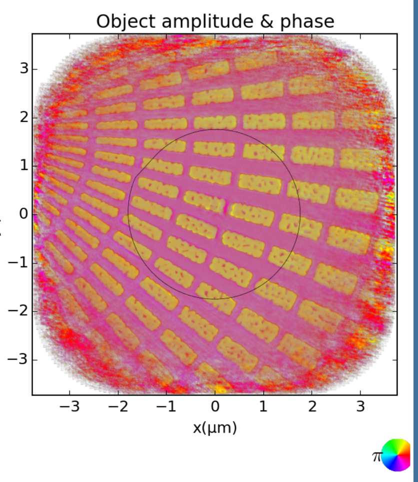
Coherent Diffraction Imaging experiments will become much more intense/faster after the EBS upgrade. The ESRF has started an open-source scientific software programme, to improve the ability of users to quickly analyse data for the various techniques, notably on the coherent imaging beamlines – to encourage users interested primarily in the results on materials rather than in the development of the technique. PyNX is a high-performance (based on GPU computing) modular toolkit which can be used for Coherent Diffraction Imaging, Ptychography and (in development) Holo-Tomography.

Main PyNX features

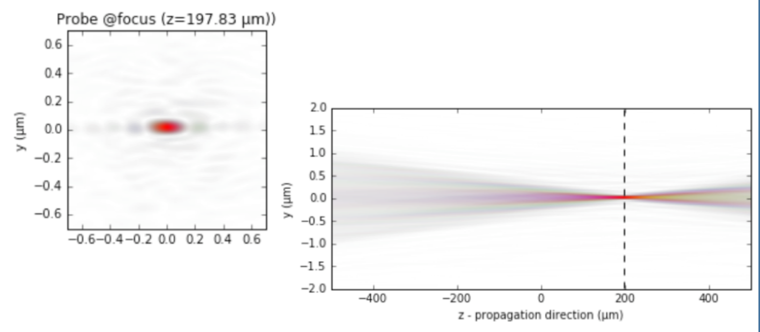
All algorithms fully executed on GPU for performance
Use *normalized log-likelihood* as a systematic figure of merit

2D Ptychography (far field & near field, Bragg):

- **Combination of algorithms:**
  - Difference Map
  - Alternating projections
  - Conjugate Gradient Maximum Likelihood
  - Incoherent background



- **Use and create CXI format datasets**
  - http://cxidb.org/cxi.html

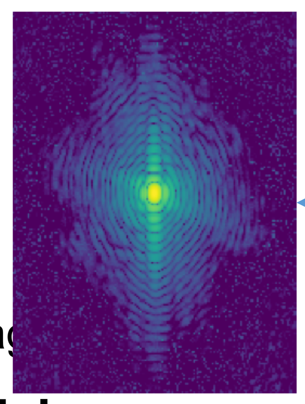


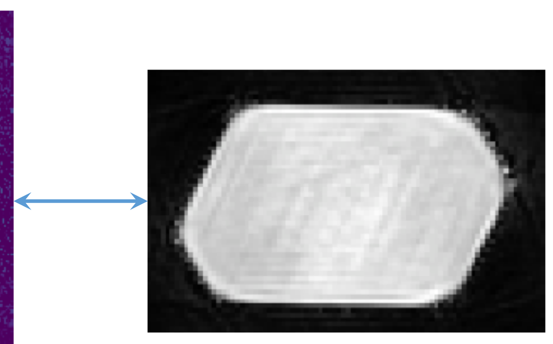
Coherent Diffraction Imaging:

- **Algorithms (2D, 3D)**:
  - HIO, ER, CF…
  - Maximum Likelihood
  - Partial coherence
  - Likelihood-sorting of solutions

Wavefront propagation:

- Near field propagation
- Far field propagation
- Fractional Fourier transform propagation





Scattering from atomistic models:

- Fast calculations using CUDA or OpenCL
- 7x10^11 atoms.reflections/s using a single *Titan V* GPU (~5 Tflop/s)

Fast coherent X-ray imaging Using Operators, Python & GPUs

All coherent imaging algorithms are based on simple combinations of mathematical operations:

- **Near and far field propagation**
- **Applying constraints in real and Fourier space**

TABLE I. Summary of various algorithms.

Algorithm	Iteration $\rho^{(n+1)}=$
ER	$P_s P_m \rho^{(n)}$
SF	$R_s P_m \rho^{(n)}$
HIO	$\begin{cases} P_m \rho^{(n)}(r) & r \in S \\ (I-\beta P_m)\rho^{(n)}(r) & r \notin S \end{cases}$
DM	$\{I+\beta P_s[(1+\gamma_s)P_m-\gamma_s I]-\beta P_m[(1+\gamma_m)P_s-\gamma_m I]\}\rho^{(n)}$
ASR	$\frac{1}{2}[R_s R_m + I]\rho^{(n)}$
HPR	$\frac{1}{2}[R_s(R_m+(\beta-1)P_m)+I+(1-\beta)P_m]\rho^{(n)}$
RAAR	$[\frac{1}{2}\beta(R_s R_m+I)+(1-\beta)P_m]\rho^{(n)}$

Marchesini, S. '*A unified evaluation of iterative projection algorithms for phase retrieval*'. Review of Scientific Instruments **78** (2007), 011301

All PyNX modules are based on operators:

- Called from Python
- Executed on GPU using OpenCL or CUDA
- GPU complexity is completely hidden
- Optimal asynchronous performance

Main design choices for PyNX:

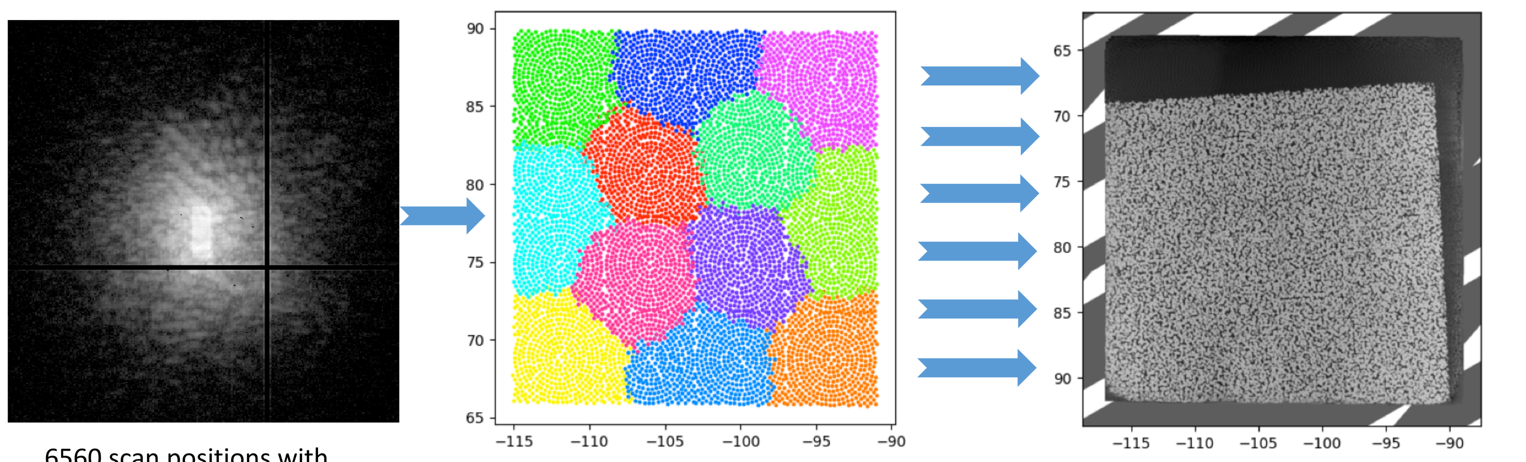
- Achieve **highest possible speed**:
  - Use GPU: ~10x more cost-efficient than CPU
- **Flexible programming**:
  - 'operator-based' approach allows fast development while retaining top performance
  - **Algorithms** can be easily tuned and without GPU programming knowledge
  - Allows different languages & GPUs (OpenCL, CUDA)

```
import numpy as np
import fabio
# This imports all necessary operators. GPU will be auto-selected
from pynx.cdi import *

iobs = np.fft.fftshift(fabio.
support = np.fft.fftshift(fabio.open("support.edf").data)
mask = np.fft.fftshift(fabio.open("mask.edf").data)

cdi = CDI(iobs, obj=None, support=support, mask=mask,wavelength=1e-10,pixel_size_detector=55e-6)
# Initial scaling, required by mask
cdi = ScaleObj(method='F') * cdi
# Do 40 cycles of Hybrid Input/Output, then 5 of ER
cdi = ER() ** 5 * HIO() ** 40 * cdi

# Support update operator
sup = SupportUpdate(threshold_relative=0.25, smooth_width=0.5, post_expand=(1,-1))

# 40 HIO + 5 ER Cycles with support update, repeated 10 times
cdi = (sup * ER() ** 5 * HIO() ** 40)**10 * cdi
```

Example code for CDI data analysis

Notebook demo movie

Distributed Ptychography analysis



6560 scan positions with nice speckles

Split 6560 scan positions in 12 areas with ~570 points (with 20-30 positions overlap)

Merge the 12 independent (fully asynchronous) reconstructions

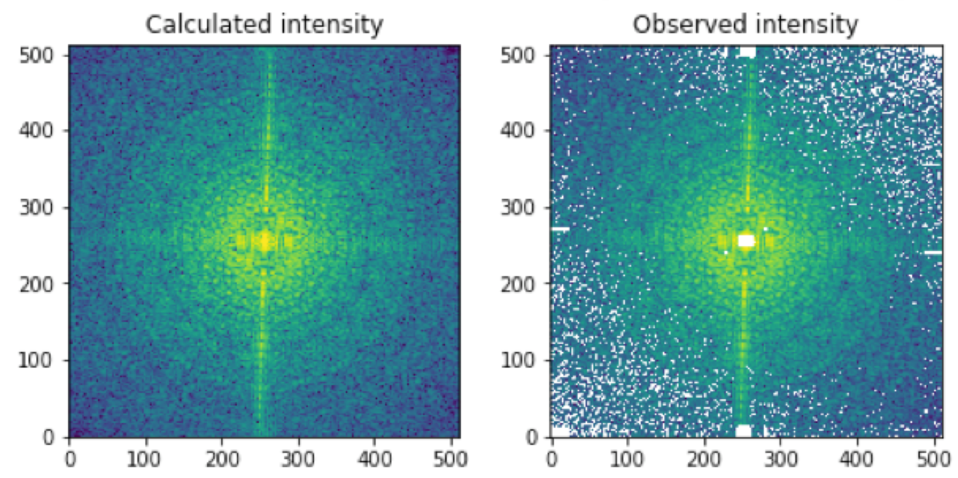
All the analysis from a simple command: (here distributed to 4 GPU)

```
mpiexec -n 4 pynx-cixpty.py mpi=split data.cxi probe=focus,60e-6x200e-6,0.116
        algorithm=ML**100,AP**1000,position=1,AP**50,AP**50,DM**200,nbprobe=3,probe=1
```

Ptychography: example speed

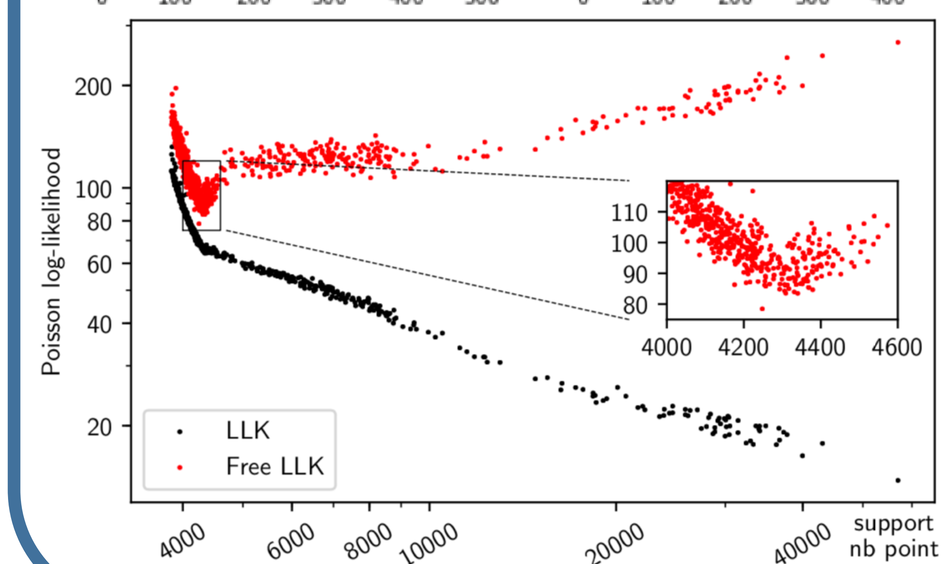
Configuration	Algorithm	Time/cycle
1000 frames, 256x256 1 mode, far field	Difference map	17 ms
	Maximum likelihood (CG)	34 ms
1000 frames, 256x256 **3 modes**, far field	Difference map	46 ms
	Maximum likelihood (CG)	93 ms
100 frames, **1024x1024** 1 mode, far field	Difference map	34 ms
	Maximum likelihood (CG)	67 ms
17 frames, 2048x2048 1 mode, **near field**	Difference map	39 ms
	Maximum likelihood (CG)	79 ms
250x10³ frames, 256x256 1 mode, far field, **12 GPU** (object 15k x 15k pixels)	Difference map	375 ms
	Maximum likelihood (CG)	760 ms

Unsupervised CDI reconstructions using free log-likelihood



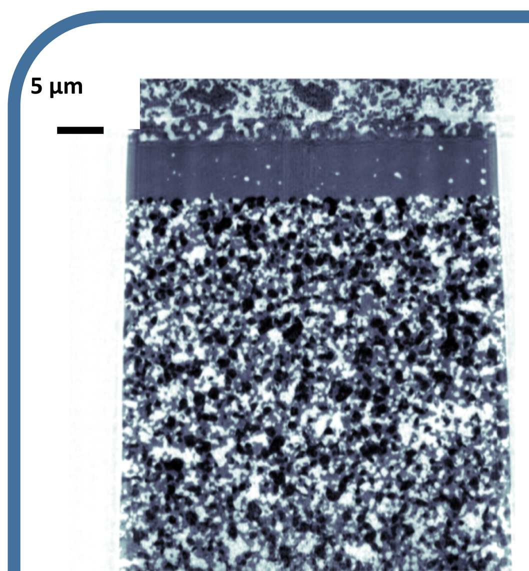
$$p(I_o|I_c) = \frac{(I_c)^{I_o}}{I_o!}e^{-I_c}$$

- Use Poisson log-likelihood (LLK) as a indicator.
- The object area (support) is unknown => ambiguity on solutions
- Solution: set aside 5% of experimental data points which are masked to the iterative algorithm => objective indicator : free log-likelihood calculated only on these pixels



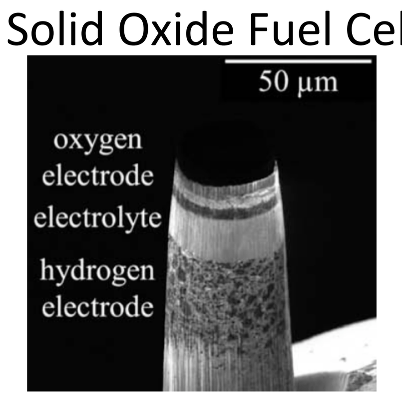
- Free log-likelihood gives an unambiguous choice for the support size, which can be automatically selected
- Example application to unsupervised, serial CDI analysis: Björling et al, J Synchrotron Rad 26 (2019), 18300

Near Field Ptychography



- Dataset: 1200 projections, 17 frames 2k x 2k
- Pixel size: 25 nm
- Volume: 50x50x50 μm3
- reconstruction: 10 min/projection
- Processing (30 P100 GPU/CEA-Idris): 7.5 h
- Data collection: 18 h (pre-EBS)

Solid Oxide Fuel Cell



Note: reconstruction is >1 order of magnitude faster for local tomography (smaller phase amplitude & gradient)

Hubert, Monaco, Da Silva, Favre-Nicolin, D. Montinaro, Cloetens, Laurencin (in preparation)
Also see: ECS Trans. **91**, 653
https://hal.archives-ouvertes.fr/hal-02279503
https://hal.archives-ouvertes.fr/hal-01526466

Holo-tomography developments

- Simultaneous reconstruction of multiple projections along with the illumination
- **Single distance fast tomography** (7s per scan, id16B):
  - 720 projections, 1280x1080 pixels, 100 nm pixel
  - PyNX phasing on 1 GPU: 2 minutes (including loading and tomography (FBP) using Nabu
- **4-distance holo-tomography**:
  - 3000 projections, 2560x2160, 27 nm pixel size
  - PyNX phasing: ~30 minutes using 6 GPUs in //
- Power9 architecture: fast main memory<->GPU transfers for large datasets

Favre-Nicolin, V. et al. PyNX: high-performance computing toolkit for coherent X-ray imaging based on operators. ***J Appl Crystallogr 53, 1404–1413 (2020)***
Favre-Nicolin, V., Leake, S. J. & Chushkin, Y. Free log-likelihood as an unbiased metric for coherent diffraction imaging. ***Scientific Reports 10, 2664 (2020).***

Acknowledgments for data sets & tests: *Steven Leake, Marie-Ingrid Richard, Jerome Carnis, Gaétan Girard (id01), Yuriy Chushkin (id10), Tilman Grünewald, Manfred Burghammer (id13), Julio Cesar da Silva, Leonid Bloch (id16a), Ross Harder (APS)*