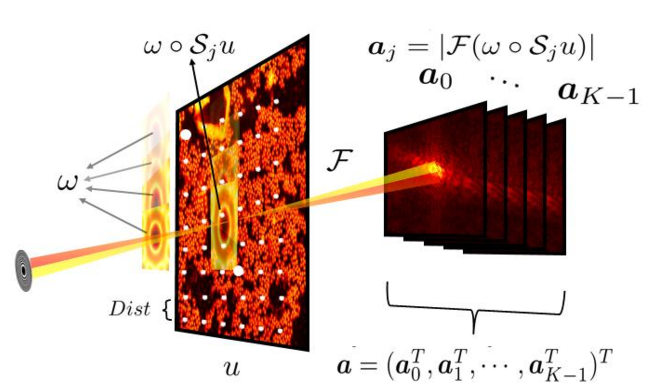
NATIONAL ACCELERATOR LABORATORY

Ptychography at extreme scales: Imaging macroscopic samples at (near) atomic resolution

By: Yuan Ni (UC Davis Math Dept.), Stefano Marchesini (SLAC)

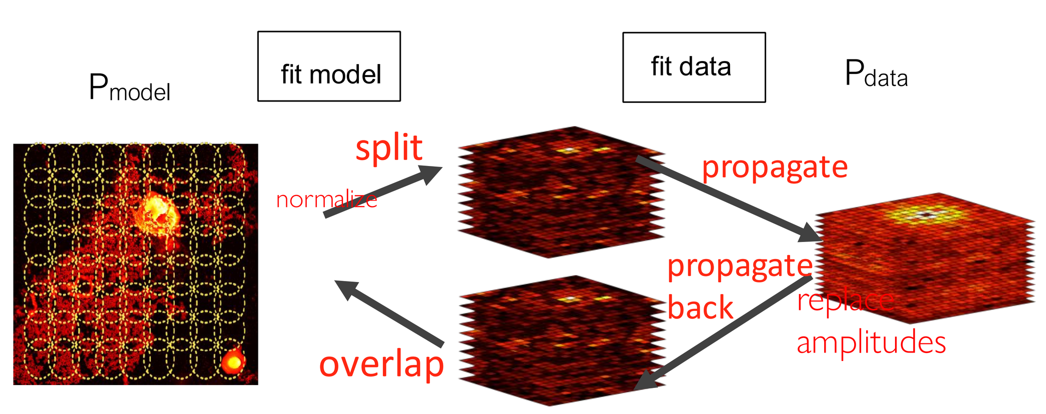
Introduction: ptychography

Ptychography is an experimental technique whereby one acquires coherent diffraction patterns from overlapping regions of a sample.



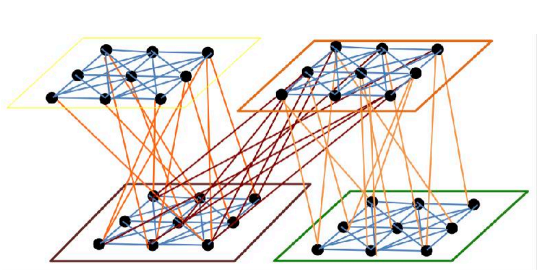
Combine: coherent diffractive imaging with scanning probe microscopy

Scattering and redundancy (provided by overlapping regions) has enabled the highest resolution x-ray and electron microscopes in the world. The typical algorithm is based on (alternating) projections (P_{model} and P_{data}):



However, at each iteration a frame communicates only with neighboring frames, therefore long range (phase) information takes many iterations to propagate. Therefore, **convergence rate drops with data size (Fig1.)**

To change this scaling behavior, we explore a graph-Laplacian technique. It relies on the relationships between overlapping pairs of frames which form a graph:



Overlapping frames are related to each other by a shift and an illumination

This technique relies on the fast computation of the **Gramian matrix** formed by a specialized **inner product between all pairs of frames**. The inner product considers the shifts among frames, the illumination and a normalization factor to account for the degree of redundancy. The high-performance operation can also (1) reduce communication in a distributed computing system and (2) help deal with slow fluctuations of experimental parameters such as drifts which are inevitable when dealing with extreme spans of length scales (from atomic to macroscopic).

Once the Gramian is computed, one can exploit an algorithm similar to Pagerank of Google fame, **enabling extreme scaling.**

The (synchronization) Optimization Problem

Solve the following optimization problem with respect to frame-wise phase:

$$\underset{\xi\in\mathbb{C}^k,|\xi|=1}{\arg\min} \; \|\underbrace{(I-P_{FQ})}_{\text{Fit model}}\underbrace{\mathrm{diag}(P_a\zeta^{(l)})}_{\text{Fit data}}B\xi\|$$

Framewise phase

Broadcast

equivalent

Find principal eigenvector

$$\underset{\omega\in\mathbb{C}^k,|\omega|=\|\mathbf{z}_{(i)}\|}{\arg\max} \; \omega^* H^{(l)}\omega \quad , \quad H^{(l)}_{i,j} := \frac{z^{*(l)}_{(i)}Q_i}{\|\mathbf{z}_{(i)}\|}\underbrace{\frac{1}{Q^*Q}}\frac{Q^*_{(j)}z^{(l)}_{(j)}}{\|\mathbf{z}_{(j)}\|}$$

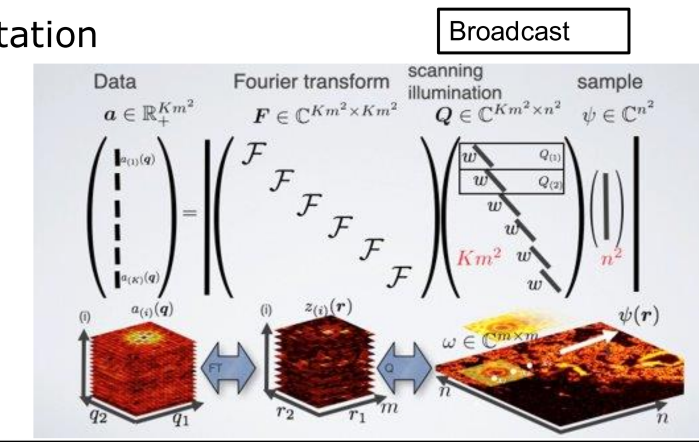
Frames after fitting the data

Adjust shift between frames and their respective illuminations

Normalize by total exposure

Normalize by total intensity

- Notation



- The k × k matrix H is computed by performing the **scalar product between every pair** of overlapping frames.
- The solution to this problem assuming constant $\|\omega\|$ is the **eigenvector** corresponding to the largest eigenvalue of the **sparse** matrix H.

Numerical Results

- Synchronization: **accelerate** convergence rate by propagating long range (phase) information. Especially with **large data.**

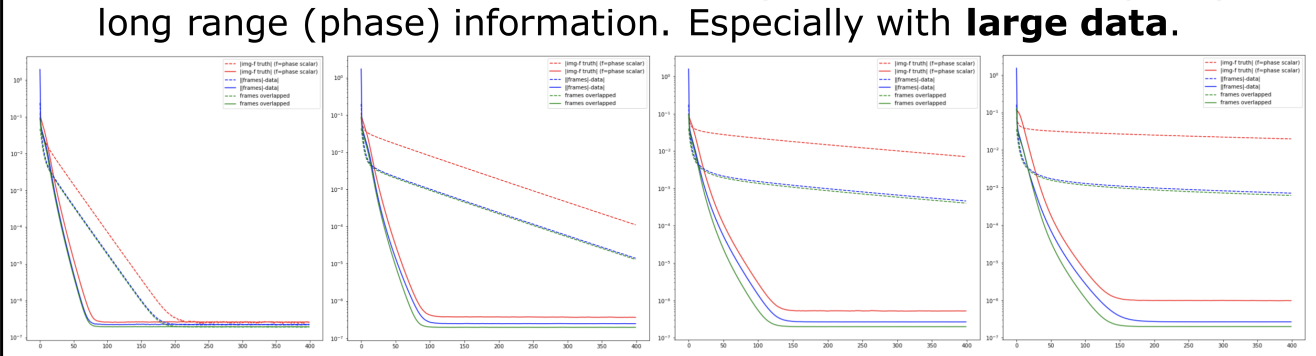


Fig 1.1 Convergence plots for different Num of frames. Left to Right Num of Frames: 8x8, 16x16,32x32,64x64

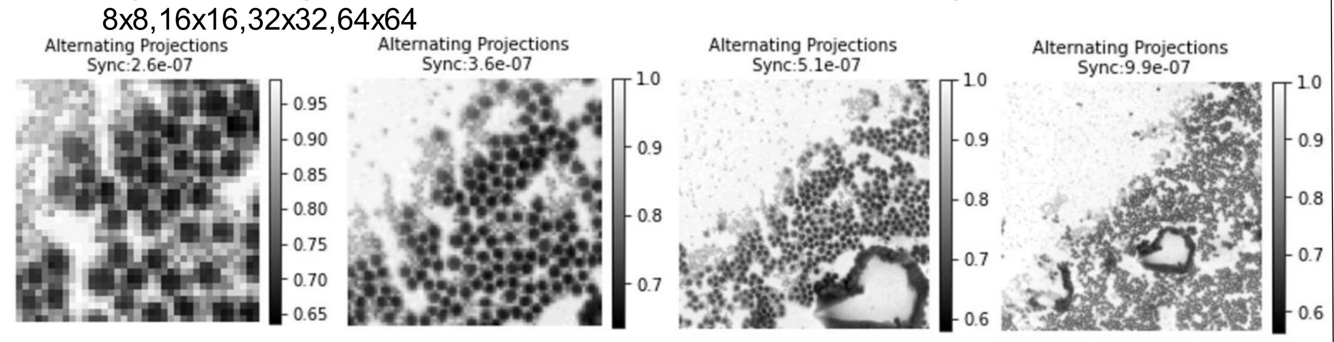


Fig 1.2 Reconstruction plots with Error |image-truth|/|truth| for different Num of frames. Left to Right Num of Frames : 8x8,16x16,32x32,64x64

- The **trade-off** between Time and Convergence rate.
  - Time/Iterations to reach the same level of accuracy.

Number of Frames	No Sync Clock Time	Iterations	Sync Clock Time	Iterations	Sync every 5 Clock Time	Iterations	Accuracy \eps_0^2
8x8	0.59s	95	0.86s	38	0.26s	58	1e-04
16x16	0.91s	407	0.94s	44	0.40s	90	1e-04
32x32	2.23s	1625	0.98s	44	0.54s	110	1e-04
64x64	7.67s	6465	0.92s	44	0.51s	120	1e-04

Note: Sync: Synchronize after every model and data fitting steps
Sync every 5: Synchronize after every 5 iterations of model and data fitting
Use 200 power iteration for Eigensolver

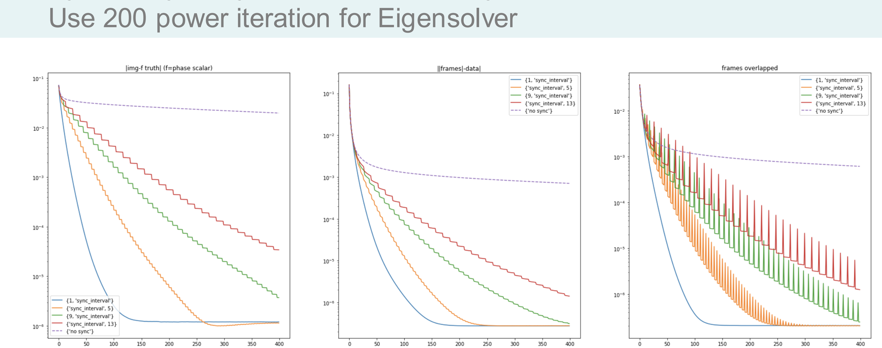


Fig 2.1 Reconstruction plots for 64x64 num of frames for different synchronization frequencies

- **Trade-off** between the Eigen-solver accuracy, Time and Convergence

# of Power Iter / # of Frames	100 % of time/ Total Time	Iter	50 % of time /Total Time	Iter	10 % of time /Total Time	Iter	1 % of time/ Total Time	Iter	Accuracy
8x8	91.7% of 0.342s	38	85.6% of 0.204s	38	58.4% of 0.095s	38	23.9% of 0.333s	56	1e-04
16x16	95.5% of 0.376s	41	85.3% of 0.224s	41	58.2% of 0.130s	51	23.3% of 0.639s	211	1e-04
32x32	91.6% of 0.388s	44	85.0% of 0.239s	45	58.3% of 0.365s	194	23.5% of 1.639s	820	1e-04
64x64	91.8% of 0.445s	49	85.7% of 1.311s	245	58.1% of 2.261s	758	22.4% of 7.04s	3240	1e-04

Note:
# of Power Iter: number of power iteration steps used to calculate the largest eigen value of H;Synchronize is applied every AP step; The frame size is fixed at 16x16 pixels. If use more balanced frame size and # of frames, the percentage of time for eigensolver would be smaller

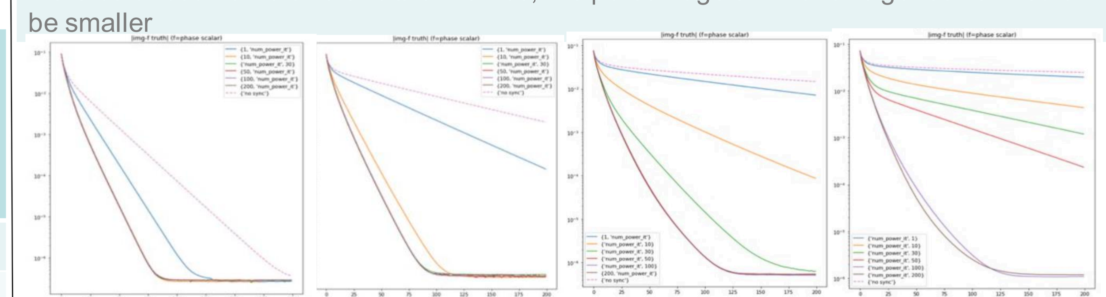
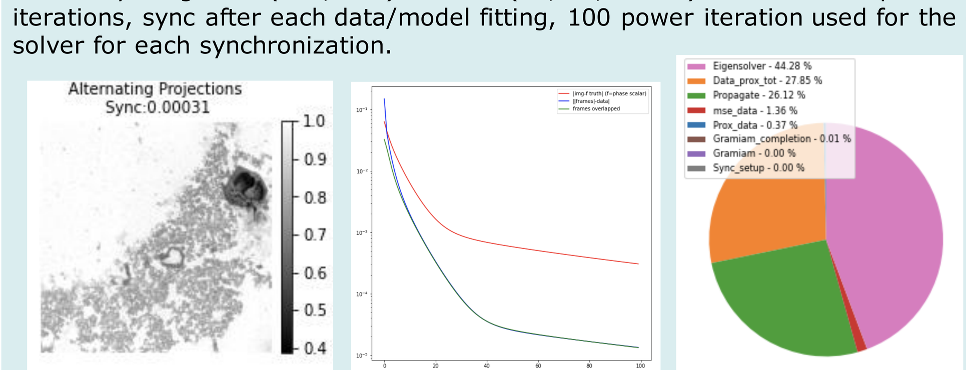


Fig 3.1 Convergence plots for different number of power iteration steps per eigen-solve . Left to Right Num of Frames: 8x8,16x16,32x32,64x64

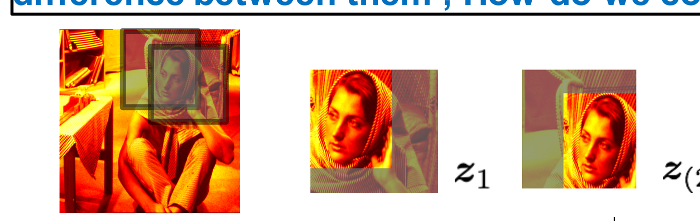
Example of an experiment output

Geometry: img size: (637, 637) frames: (32, 32, 16384). Iteration setup: 100 AP iterations, sync after each data/model fitting, 100 power iteration used for the eigen-solver for each synchronization.



Left to Right: 1. Reconstruction is visually identical to the truth. 2. Convergence Rate plot 3. Time for operations. Total Time: 4.36s

Phase synchronization:
Suppose we reconstruct frames independently; they will have a phase difference between them ; How do we correct for this?



$$z_1 = \xi_{1,2} z_2$$

Best fit $\quad \min_{|\xi|=1} \|z_{(1)} - \xi z_{(2)}\|^2$

Equivalent: $\quad \underbrace{\|z_{(1)}\|^2 + \|z_{(2)}\|^2}_{\text{constant}} - \underbrace{2\Re\left(z^*_{(2)} z_{(1)}\right)\xi^*}_{\text{maximize product}}$

Align phases: $\quad \xi = \frac{\left(z^*_{(1)} z_{(2)}\right)}{\left|\left(z^*_{(1)} z_{(2)}\right)\right|}$ Normalize the inner product

For many frames:

$$\sum_{i,j}\|\xi_{(i)} z_{(i)} - \xi_{(i)} z_{(j)}\|^2 = \sum_{i,j}\overbrace{\|z_i\|^2 + \|z_i\|^2}^{\text{constant}} - 2\xi^*_{(i)}\mathcal{H}_{(i,j)}\xi_{(j)},$$

Gramian matrix: $\quad \mathcal{H}_{(i,j)} = \langle z_i z_j \rangle$

Find largest eigenvalue: $\quad \max_\xi \xi^* (\mathcal{H})\, \xi$

Flowchart for CUDA Kernel: Gramian_calculator

Initialization:

Each block handles a pair of overlapping frames

Each thread handles pairs of overlapping pixels of the frames

Parallel Loop through Overlapping Pixels within a pair of frames:

Calculate offset

For each overlapping pixel within the integration width and height:

Sum up the value to the dot product based on frames, illumination, and normalization data.

Block-Wise Summation: CUB BlockReduce to compute the block-wide sum

Hermitian Property Handling:

For entries for frames overlapping with itself, set the sum to be real value

Fill values to the upper triangular part of H

Output Assignment:

Store the sums in global memory

Use the Hermitian property to fill the lower triangular part of H

Conclusions

Conclusion and future work

1. Developed cuda kernel to achieve fast computation of the Gramian, which was the most time-consuming part of the synchronization strategy
2. Tested different eigen-solvers and synchronization frequency
3. Achieved improved scaling performance over large datasets
4. For high level of noise, long range phase information (~100x larger than the probe) is lost regardless of the algorithm.

Future work: The high-performance kernel can also

1. Reduce communication in a distributed computing system.
2. Deal with slow fluctuations of experimental parameters such as drifts which are inevitable when dealing with extreme spans of length scales

Acknowledgments

Special thanks to my mentor Stefano Marchesini.

LCLS, SLAC, U.S. DOE, NSF GRFP.,

Reference: Marchesini, S., Schirotzek, A., Yang, C., Wu, H., & Maia, F. (2013). Augmented projections for ptychographic imaging. *Inverse Problems*, *29*(11), 115009. https://doi.org/10.1088/0266-5611/29/11/115009