Inverse Modeling Nanostructures from X-Ray Scattering Data through Massive Parallelism

Abhinav Sarje **Xiaoye S Li** **Slim Chourou** **Dinesh Kumar** **Singanallur Venkatakrishnan** **Alexander Hexemer**

Computational Research Division

Advanced Light Source

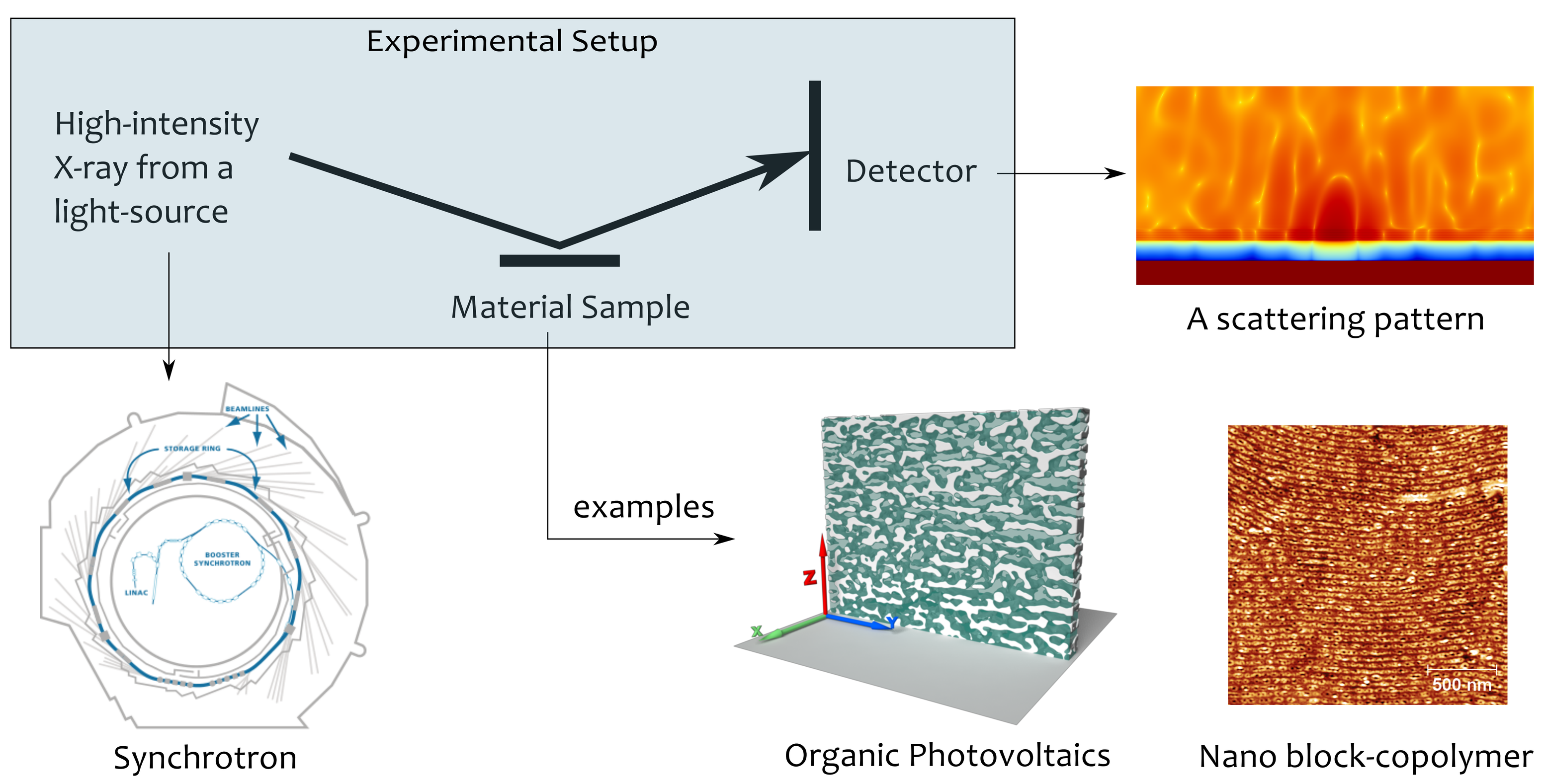
Lawrence Berkeley National Laboratory

Abstract

We consider the problem of reconstructing material nanostructures from grazing-incidence small-angle X-ray scattering (GISAXS) data obtained through experiments at synchrotron light-sources. This is an important tool for characterization of macromolecules and nano-particle systems applicable to applications such as design of energy-relevant nano-devices. Computational analysis of experimentally collected scattering data has been the primary bottleneck in this process. We exploit the availability of massive parallelism in leadership-class supercomputers with multi-core and graphics processors to realize the compute-intensive reconstruction process. To develop a solution, we employ various optimization algorithms including gradient-based LMVM, derivative-free trust region-based POUNDerS, and particle swarm optimization, and apply these in a massively parallel fashion. We compare their performance in terms of both quality of solution and computational speed. We demonstrate the effective utilization of up to 8,000 GPU nodes of the Titan supercomputer for inverse modeling of organic-photovoltaics (OPVs) in less than 15 minutes.

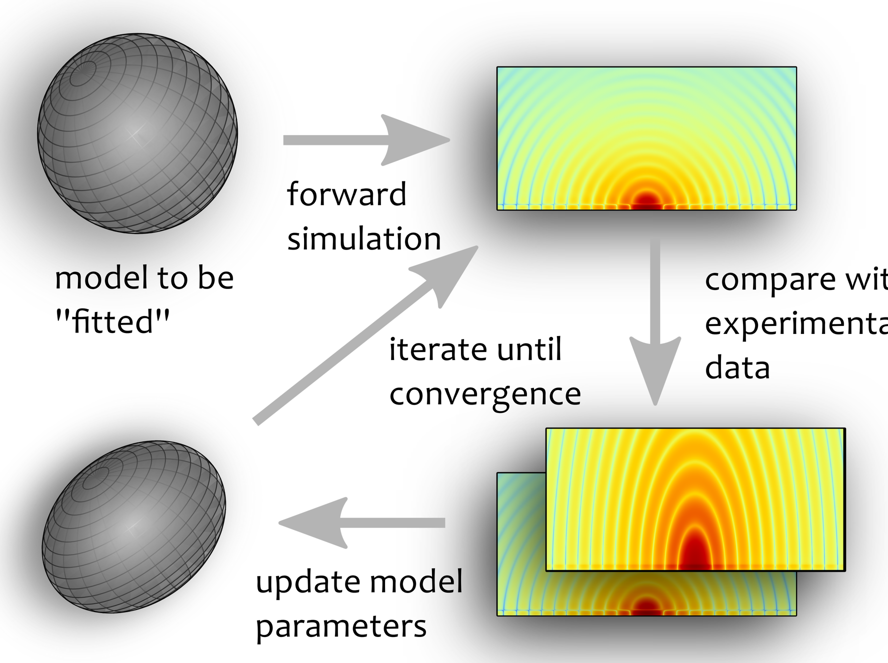
X-Ray Scattering

X-ray scattering comes in various flavors, and here we focus on data obtained using methods from the class of **grazing incidence X-ray scattering**, such as GISAXS (small-angle) and GIWAXS (wide-angle).



Inverse Modeling

The process of recovering physical properties using experimentally measured data is commonly known as inverse modeling and appears in numerous applications. This generally involves minimization of an objective function value iteratively until convergence w.r.t. a set of parameters to be "fitted", or recovered. In our case, this function is a computationally intensive forward simulation of the scattering patterns. This forward simulation is based on the Distorted Wave Born Approximation (DWBA) theory and is the most computationally intensive component. We have developed and implemented it [1, 2, 3] as a massively parallel high-perfomance code utilizing multi-core CPUs and Nvidia graphics processors (GPUs).

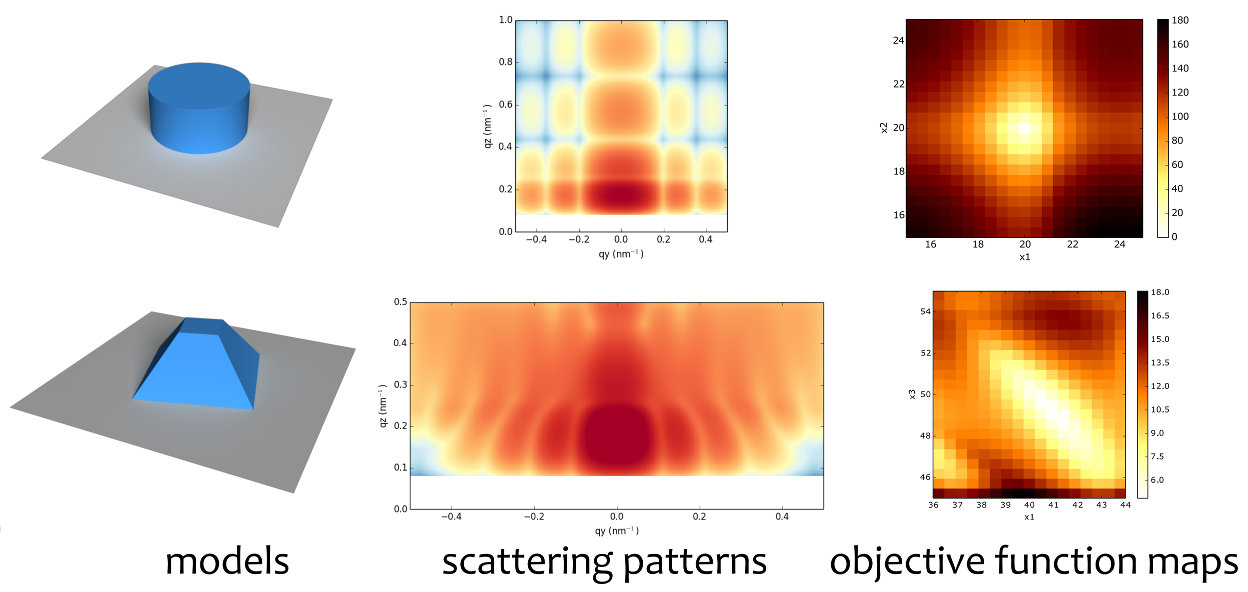


Solutions

We exploit massive parallelism to develop solutions to the problem at hand. In order to perform the inverse modeling we explore optimization algorithms including the gradient-based Limited Memory Variable Metric (LMVM), derivative-free trust-region-based POUNDerS, and stochastic Particle Swarm Optimization (PSO). LMVM and POUNDerS implementations available in the TAO optimization package are used, and we develop a parallel implementation of PSO.

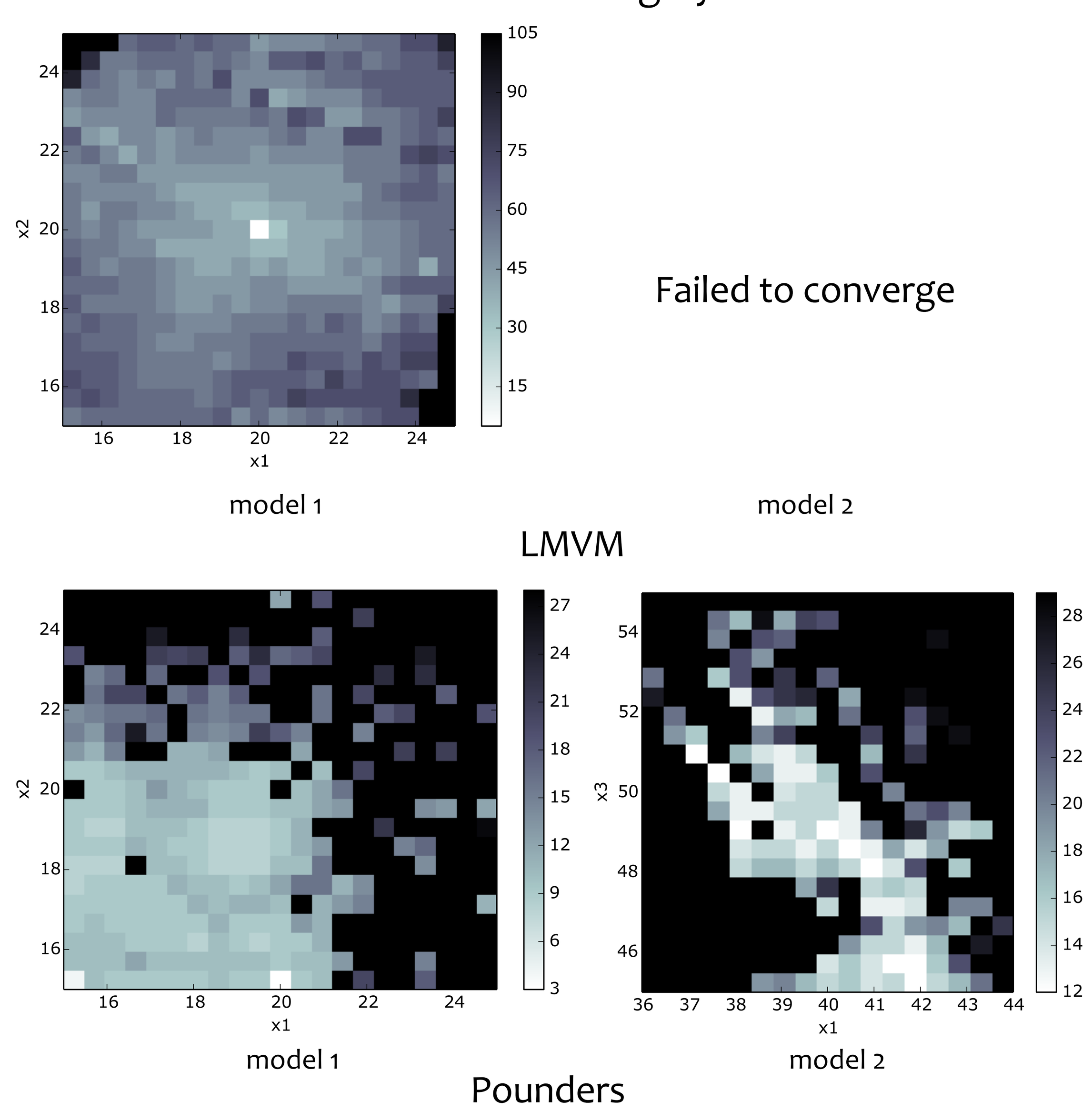
In order to evaluate the convergence performance, we use two simulated models:

1. Cylindrial nanoparticle (2 parameters)
2. Lattice of pyramidal nanoparticles (6 parameters)



Inverse Modeling with LMVM and POUNDerS

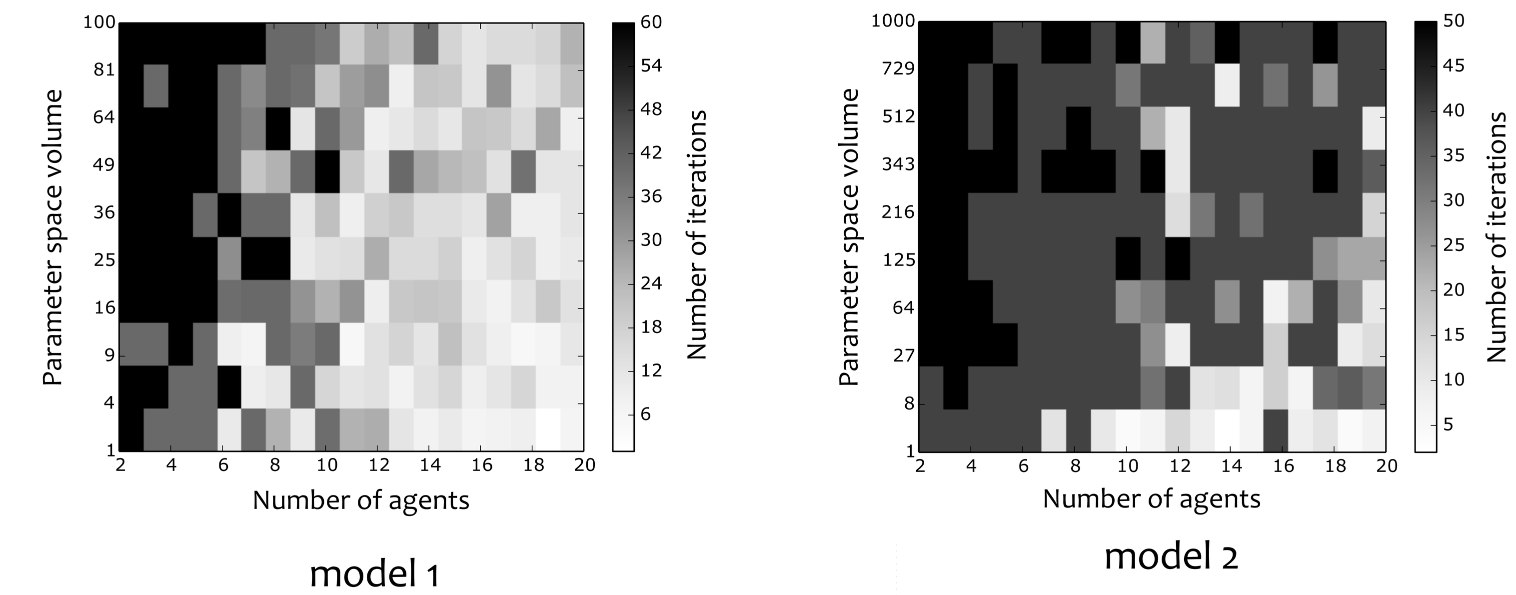
Shown below are the convergence maps for the two samples described above obtained with LMVM and POUNDerS, respectively. LMVM is able to achieve a high convergence rate for model 1, which has a single minimum as seen in the objective function map above, but it fails to converge for model 2, which has multiple minima. Comparatively, POUNDerS is more robust as it is able to converge in both cases, although the conergence rate is less than 30% in the latter case. This shows that it is highly sensitive to the initial conditions.



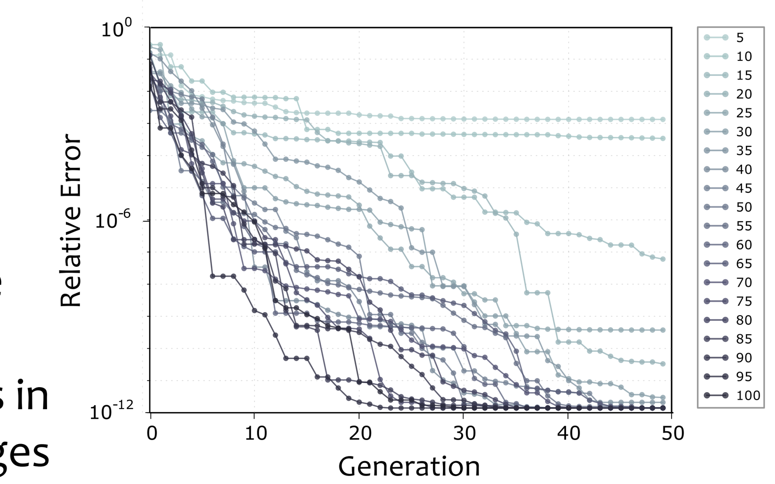
(Left) Convergence maps obtained with LMVM (top) and POUNDerS (bottom). Black areas show that convergence was not reached. Color scale shows the number of objective function evaluations required to reach convergence. On average for the first model, LMVM required about 3 times more function evaluations than POUNDerS.

Inverse Modeling with PSO

The results for convergence for the two models are shown below. These are shown with respect to varying number of agents, and varying volume of the search space. Black spots represent those configurations which failed to converge in one trial.

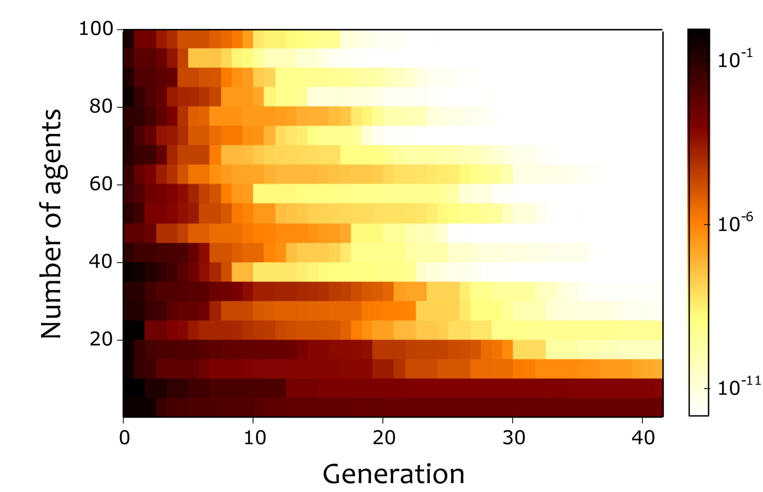


What is better for convergence: fewer agents run for a long time, or, a larger number of agents run for short time?



The plots on the right demonstrate the answer to this question. Each run was performed on a total of 100 GPU nodes in parallel. There are three main advantages of having large number of agents:

1. **Guarantee**: It provides a higher probability of reaching convergence.
2. **Efficiency**: Available massive parallelism can be effectively exploited.
3. **Speed**: Convergence is reached in fewer generations, i.e. faster time to solution.



We also demonstrate the computational speed on the Titan supercomputer with real data obtained through X-ray scattering experiments on Organic Photovoltaics (OPVs). The fitting performance results with 20 iterations are shown below.

# GPU Nodes	# Agents	Total Time	Time/agent/iteration	Relative Speedup
500	20	3110.0 s	7.775 s	1
2000	50	2071.6 s	2.071 s	3.754
8000	80	865.6 s	0.541 s	14.372

Conclusions

For the problem of recovering nanostructures from X-ray scattering patterns, the PSO method has proven to be highly effective. Although requiring high number of function evaluations, this method has a high convergence rate, reaching near perfect given enough number of agents. This works to benefit with parallel computing since PSO is highly concurrent in nature and massive parallelism can be easily exploited to bring down the actual computational time. Although the LMVM and POUNDerS methods perform well with small number of parameters, they, LMVM in particular, are less effective for higher number. They are also highly sensitive to the intial parameter value guesses. Compared to these two methods, PSO also has an additional advantage of not requiring an initial guess.

Acknowledgements

The authors thank Nvidia for providing us with several GPU cards which were used as the primary development platform for this work. This work was supported by the Director, Office of Science, U.S. Department of Energy (DoE) under contract no. DE-AC02-05CH11231. This work was also supported by DoE Early Career Research Program grant awarded to Alexander Hexemer. This research used resources of the National Energy Research Scientic Computing Center (NERSC), which is supported by the Office of Science, U.S. DoE, under contract no. DE-AC02-05CH11231, and of the Oak Ridge Leadership Computing Facility (OLCF), which is supported by the Office of Science, U.S. DoE, under contract no. DE-AC05-00OR22725.

References

1. S Chourou, A Sarje, X Li, E Chan, and A Hexemer. **HipGISAXS: A High Performance Computing Code for Simulating Grazing Incidence X-Ray Scattering Data**. Journal of Applied Crystallography, vol 46, no 6, pp 1781-1795, 2013.
2. A Sarje, X Li, S Chourou, E Chan, and A Hexemer. **Massively Parallel X-ray Scattering Simulations**. In International Conference for High Performance Computing, Networking, Storage and Analysis (Supercomputing), pp 46:1-46:11, 2012.
3. A Sarje, X Li, and A Hexemer. **Tuning HipGISAXS on Multi and Many Core Supercomputers**. In Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems, 2013.