



Condor: a simulation tool for flash X-ray imaging¹

 Max F. Hantke,^{a,*} Tomas Ekeberg^a and Filipe R. N. C. Maia^{a,b}

^aLaboratory of Molecular Biophysics, Department of Cell and Molecular Biology, Uppsala University, Husargatan 3 (Box 596), SE-751 24 Uppsala, Sweden, and ^bNERSC, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA. *Correspondence e-mail: hantke@xray.bmc.uu.se

Received 3 February 2016

Accepted 7 June 2016

Edited by N. D. Loh, National University of Singapore

¹This article will form part of a virtual special issue of the journal on free-electron laser software.

Keywords: femtosecond coherent diffractive imaging; X-ray free-electron lasers; simulation; single-particle imaging; computer programs.

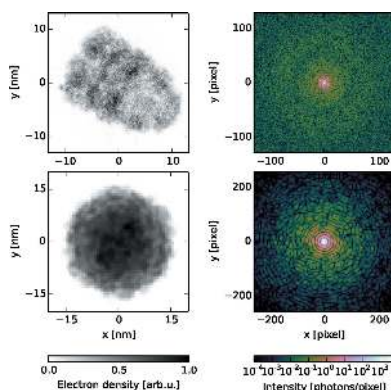
Flash X-ray imaging has the potential to determine structures down to molecular resolution without the need for crystallization. The ability to accurately predict the diffraction signal and to identify the optimal experimental configuration within the limits of the instrument is important for successful data collection. This article introduces *Condor*, an open-source simulation tool to predict X-ray far-field scattering amplitudes of isolated particles for customized experimental designs and samples, which the user defines by an atomic or a refractive index model. The software enables researchers to test whether their envisaged imaging experiment is feasible, and to optimize critical parameters for reaching the best possible result. It also aims to support researchers who intend to create or advance reconstruction algorithms by simulating realistic test data. *Condor* is designed to be easy to use and can be either installed as a Python package or used from its web interface (<http://lmb.icm.uu.se/condor>). X-ray free-electron lasers have high running costs and beam time at these facilities is precious. Data quality can be substantially improved by using simulations to guide the experimental design and simplify data analysis.

1. Introduction

Flash X-ray imaging (FXI) may become a tool to solve structures down to molecular resolution without the need for crystallization (Neutze *et al.*, 2000; Bergh *et al.*, 2008). By employing femtosecond pulses produced by X-ray free-electron lasers, FXI can outrun radiation damage processes that limit resolution (Chapman *et al.*, 2006). FXI dispenses with image forming lenses and thereby circumvents the difficulty of manufacturing efficient lenses for X-rays (Chapman & Nugent, 2010). Aerosol sample delivery avoids a sample support, which means that the structure can be imaged with practically no background (Bogan *et al.*, 2008; Seibert *et al.*, 2011; Hantke *et al.*, 2014).

For reaching the goal of 3 Å resolution, the Single Particle Imaging Initiative identifies the requirement of simulations that realistically represent the experiment conditions to guide future development (Aquila *et al.*, 2015). It is essential to optimize and harmonize all relevant experimental parameters, such as photon wavelength, photon flux, illumination profile, camera distance, detector settings, sample density and even sample type. Being able to accurately predict diffraction data facilitates optimization of the experimental setup and helps to provide accurate estimates of the expected data quality. Simulation tools can help researchers to use their beam time more efficiently and measure diffraction data at the highest possible quality.

Software for simulating X-ray diffraction data exists. For crystal diffraction, for example, *CCP4* (Winn *et al.*, 2011) is widely used. But it is aimed at crystal diffraction, making it



OPEN ACCESS

not carry any structural information and is confined to the forward direction. It usually passes through a gap between the detector panels or is blocked by a beam stop and is never measured. Structural information about the sample is encoded by the scattering amplitude $\Psi^{(1)}$, which is the superposition of spherical waves with amplitude $\varphi(\mathbf{x})$ originating from all points \mathbf{x} in the scattering volume:

$$\Psi^{(1)}(\mathbf{x}') = \iiint \frac{\exp(ik|\mathbf{x}' - \mathbf{x}|)}{|\mathbf{x}' - \mathbf{x}|} \varphi(\mathbf{x}) \exp(i\mathbf{k}_0 \cdot \mathbf{x}) \, d\mathbf{x}. \quad (1)$$

In our scenario, the sample volume is small and the detector distance large. Hence, we may safely assume $|\mathbf{x}'| \gg |\mathbf{x}|$ and obtain the far-field approximation of (1):

$$\Psi^{(1)}(\mathbf{q}) = \frac{\exp(ikr)}{r} \iiint \varphi(\mathbf{x}) \exp(-i\mathbf{q} \cdot \mathbf{x}) \, d\mathbf{x}, \quad (2)$$

where $\mathbf{q} = \mathbf{k}_1 - \mathbf{k}_0$ denotes the scattering vector and $r = |\mathbf{x}'|$. Since we only consider elastic scattering the energy is conserved and so is the wavenumber $k = |\mathbf{k}_0| = |\mathbf{k}_1| = 2\pi/\lambda$, where λ denotes the wavelength. As we are only interested in relative phase differences we neglect the phase factor $\exp(ikr)$ in the following equations.

For numerical calculation of the scattering amplitude $\Psi^{(1)}(\mathbf{q})$ we have to either solve the integral in (2) or approximate it by a discrete function. Analytical solutions exist for certain sample models, such as uniformly filled spheres or spheroids (Feigin & Svergun, 1987; Hamzeh & Bragg, 1974). In *Condor* these solutions of (2) are implemented and can be customized by a few parameters. For more complex samples *Condor* provides two ways of defining the sample: either by a positional arrangement of atoms or by a gridded refractive index map. In the following subsections numerical solutions for these two particle models are presented. Both involve approximating the integral in (2) by discrete Fourier transforms (DFTs) that have the general form

$$\hat{F}(\mathbf{s}) = \frac{1}{N} \sum_{j=0}^{N-1} F(\mathbf{x}_j) \exp\left(-2\pi i \frac{\mathbf{s} \cdot \mathbf{x}_j}{N}\right). \quad (3)$$

This formulation allows *Condor* to deploy efficient fast Fourier transform algorithms and exploit rapid parallel computing architectures.

2.1. Atomic model

FXI studies often target small sample particles that have sufficient resemblance to systems for which atomic structures have been determined by either X-ray crystallography, cryo-electron microscopy or nuclear magnetic resonance spectroscopy. X-rays are scattered by atoms because of their bound electrons. The scattering strength of a single free electron is known as the Thomson scattering length r_0 . The scattering potential for N free electrons located at the respective positions \mathbf{x}_j may be written as

$$\varphi_{\text{electrons}}(\mathbf{x}) = \sum_{j=1}^N \delta(\mathbf{x} - \mathbf{x}_j) r_0. \quad (4)$$

Table 1

Mass density, atomic composition and refractive index for a selection of optical media.

The material constants were taken from Bergh *et al.* (2008), Molla *et al.* (1991) and Dans *et al.* (1966), and the refractive index was calculated from these values with the relationship given by (10).

Material type	Density (g cm ⁻³)	Atomic composition	Refractive index ($\lambda = 1.240$ nm)
Water	1.00	H ₂ O	$1 - (1.54 + 0.18i) \times 10^{-4}$
Protein	1.35	H ₈₆ C ₅₂ N ₁₃ O ₁₅ S	$1 - (2.03 + 0.16i) \times 10^{-4}$
DNA	1.70	H ₁₁ C ₁₀ N ₄ O ₆ P	$1 - (2.44 + 0.23i) \times 10^{-4}$
Lipid	1.00	H ₆₉ C ₃₆ O ₆ P	$1 - (1.54 + 0.10i) \times 10^{-4}$
Cell	1.00	H ₂₃ C ₃ N ₁₀ S	$1 - (1.51 + 0.16i) \times 10^{-4}$
Poliovirus particle	1.34	C ₃₃₂₆₅₂ H ₄₉₂₃₈₈ N ₉₈₂₄₅ ⁻ O ₁₃₁₁₉₆ P ₇₅₀₁ S ₂₃₄₀	$1 - (1.99 + 0.17i) \times 10^{-4}$

By substituting (4) into (2) the δ functions conveniently reduce the integral in (2) to a sum and we obtain the scattering amplitude in a simpler form:

$$\Psi_{\text{electrons}}^{(1)}(\mathbf{q}) = r^{-1} \sum_{j=0}^N r_0 \exp(-i\mathbf{q} \cdot \mathbf{x}). \quad (5)$$

For electrons bound to an atom of species a the scattering length can be calculated by multiplying r_0 with the atomic scattering factor $f_a^{(\lambda)}(\theta)$. The atomic scattering factor is a semi-empirically determined element-specific constant that is tabulated for a large range of wavelengths λ and scattering angles θ (Brown *et al.*, 2006; Henke *et al.*, 1993). The shape of the atom is reflected in the angular dependency; hence the atomic scattering factor is also known as the atomic form factor.

This permits us to replace the integral in (2) as in (5) by a sum. The scattering amplitude can be evaluated by separating the calculation into sums for each atom species a that accounts for N_a atoms at positions $\{\mathbf{x}_j^{(a)}\}$. We obtain

$$\Psi_{\text{atoms}}^{(1)}(\mathbf{q}) = r^{-1} \sum_a f_a^{(\lambda)}(\theta) \left[\sum_{j=1}^{N_a} r_0 \exp(-i\mathbf{q} \cdot \mathbf{x}_j^{(a)}) \right]. \quad (6)$$

$\Psi_{\text{atoms}}^{(1)}(\mathbf{q})$ now has the form of a sum of DFTs (3) with $F(\mathbf{x}_j) = r_0$ computed on the nonregular grid $\{\mathbf{x}_j^{(a)}\}$.

2.2. Refractive index model

For larger objects, such as big protein complexes or virus particles, the atomic structure is rarely on hand. However, at lower than atomic resolution electron density maps $\rho_e(\mathbf{x})$ of a wide range of structures have been measured by electron microscopy. Also, for many relevant optical media we can estimate the atomic composition (see Table 1) and are able to model samples by customized density maps of optical media. For these cases the scattering potential $\varphi_n(\mathbf{x})$ can be derived from the Maxwell equations and written as a function of the complex valued refractive index $n(\mathbf{x})$:

$$\varphi_n(\mathbf{q}) = \frac{2\pi}{\lambda^2} [1 - n(\mathbf{x})]. \quad (7)$$

For convenience we define $\partial n(\mathbf{x}) \equiv 1 - n(\mathbf{x})$. By inserting (7) into (2) we obtain the scattering amplitude as a function of ∂n :

$$\Psi_{\partial n}^{(1)}(\mathbf{q}) = r^{-1} \iiint \frac{2\pi}{\lambda^2} \partial n(\mathbf{x}) \exp(-i\mathbf{q} \cdot \mathbf{x}) \, d\mathbf{x}. \quad (8)$$

If this equation is interpreted as the continuum limit of (5) the relationship between the refractive index and the electron density distribution $\rho_e(\mathbf{x})$ becomes

$$n_{\text{el}}(\mathbf{x}) = 1 - \frac{\lambda^2}{2\pi} r_0 \rho_{\text{el}}(\mathbf{x}), \quad (9)$$

and the relationship between refractive index and the atom density distribution $\rho_a(\mathbf{x})$ becomes

$$n_{\text{at}}(\mathbf{x}) = 1 - \frac{\lambda^2}{2\pi} r_0 \sum_{a=0}^M f_a^{(\lambda)}(0) \rho_a(\mathbf{x}). \quad (10)$$

Using the relationships (9) and (10), *Condor* converts electron and atom density maps into refractive index maps. We presume here that $f_a^{(\lambda)}(\theta) \simeq f_a^{(\lambda)}(0)$ for all scattering angles θ , which is a valid assumption if the resolution of the measurement is well below atomic length scales.

Discretization of the Fourier integral in (2) with (3) on a three-dimensional cubic grid of $L \times L \times L$ points at spacing Δx results in

$$\Psi_n^{(1)}(\mathbf{q}, r) = r^{-1} \frac{2\pi}{\lambda^2} L^3 \widehat{\partial n} \left(\frac{\mathbf{q}\Delta x}{2\pi} \right) \Delta x^3. \quad (11)$$

with $\widehat{\partial n}$ being the Fourier transform of ∂n . This expression allows *Condor* to efficiently calculate the scattering amplitude for any discrete map $\{\partial n(\mathbf{x}_j)\}$ on the regular grid $\{\mathbf{x}_j\}$.

2.3. Diffraction measurement

To predict the absolute scattering signal $I^{(1)}(\mathbf{q})$ measured with a photon detector we need to take into account the intensity I_0 of the illumination, the solid angle $\Omega(\theta)$ that is covered by the detector pixel, and the polarization factor $P(\theta)$, which accounts for the effects of the polarization of the incoming beam in the scattered signal (Als-Nielsen & McMorrow, 2001). With these parameters the expectation value for the number of scattered photons measured in a pixel (without noise and any losses) is given by

$$I^{(1)}(\mathbf{q}) = I_0 |\Psi^{(1)}(\mathbf{q})|^2 P(\theta) \Omega(\theta). \quad (12)$$

Owing to the quantum nature of photons the measurement of $I^{(1)}(\mathbf{q})$ inevitably suffers from shot noise and thus follows Poisson statistics. This type and other types of measurement errors such as detector noise, parasitic scattering and limited quantum efficiency may be added to the simulated intensity values if desired.

For the refractive index model the agreement of data from a real FXI experiment and simulated data calculated by using the formalism that has been described here is demonstrated in Fig. 3. For the atomic model such a comparison cannot be made because we lack suitable experimental data at this point.

3. Usage

In the following paragraphs we give an introduction to the usage and functionality of *Condor*. For a detailed description of all features please see *Condor*'s documentation at <http://lmb.icm.uu.se/condor/documentation>.

Every *Condor* simulation requires the configuration of at least three components: the X-ray source, at least one sample and a pixel array detector. The configuration of the X-ray source defines the photon wavelength and intensity at the interaction point. The model of the sample can be of different kinds, either an atomic model or a refractive index description. The atomic description requires knowledge about all atom positions and atom species in the scattering volume. For example the online Protein Data Bank (PDB; Berman *et al.*, 2000) is a resource that provides a wide range of structures at atomic resolution. The structure can be provided either by a list of coordinates and atomic numbers or by a PDB file or PDB ID code.

To define a refractive index map *Condor* accepts a three-dimensional array of data points on a cubic Cartesian grid or the geometrical parameters of a sphere or spheroid. The map values can be refractive indices, electron densities or atom densities. For the last two, formulas (9) or (10) are used for the conversion to refractive indices. *Condor* interfaces to the Electron Microscopy Databank (EMDB; Lawson *et al.*, 2011), from which density maps can be retrieved. The orientation of the particle is defined by an extrinsic rotation. The rotation can be defined by either a triple of Euler angles, a rotation matrix or a quaternion. Multiple particles at different positions in the beam can be simulated as well. The configuration of the pixel detector determines the position of all pixels in

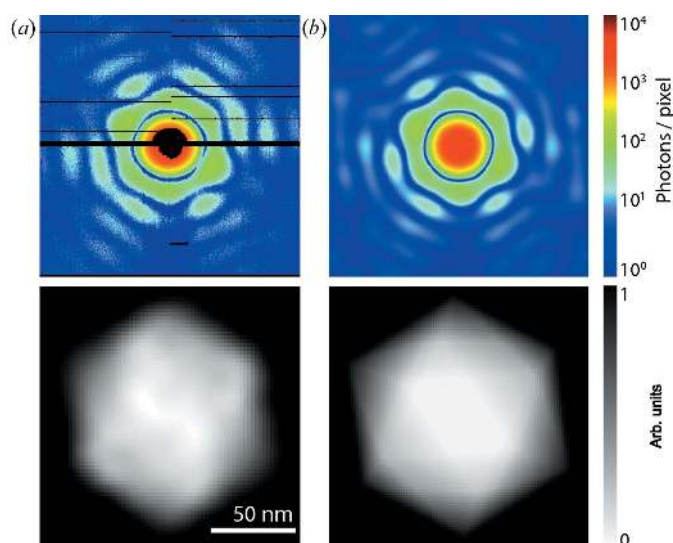


Figure 3 Comparison of experimental data and simulated data. (a) From the measured FXI diffraction pattern (top) of a single carboxysome (*i.e.* an icosahedral cell organelle) the projection image (bottom) was reconstructed by iterative phase retrieval down to 18.1 nm resolution. (b) At the given resolution the *Condor* simulation of a diffraction pattern and projection image for a uniformly filled icosahedron in matching orientation and size provides an acceptable approximation for the data shown in (a). Figure adapted from Hantke *et al.* (2014).

computer programs

space with respect to the interaction point. The detector noise, the fluctuating beam tilt, the saturation level, a missing data mask *etc.* may also be specified.

The default way of carrying out a *Condor* simulation is by calling the executable *condor* from a folder that contains a configuration file named *condor.conf*. Fig. 4 shows two

(a)

```
[source]
# Photon
wavelength = 1E-10
# Pulse
pulse_energy = 1E-3
# Focus profile
focus_diameter = 0.1E-6

[particle_atoms_PDB1AON]
# PDB entry 1AON (GROEL/GROES)
pdb_filename = 1AON.pdb
# Orientation of particle
rotation_formalism = quaternion
rotation_values = [0.58, 0.13, -0.77, 0.23]

[detector]
# Geometry
distance = 0.1
pixel_size = 300E-6
# Number of pixels
nx = 256
ny = 256
# Center variation
center_variation = uniform
center_spread_x = 5
center_spread_y = 9
```

(b)

```
[source]
# Photon
wavelength = 1E-10
# Pulse
pulse_energy = 1E-3
# Focus profile
focus_diameter = 0.1E-6

[particle_map_EMD1144]
# Map derived from
# EMD entry 1144 (Poliovirus)
geometry = custom
map3d_filename = emd_1144.h5
map3d_dataset = dn_map
dx = 2.69E-10
# Orientation of particle
rotation_formalism = quaternion
rotation_values = [0.01, 0.69, 0.69, -0.22]

[detector]
# Geometry
distance = 0.4
pixel_size = 125E-6
# Number of pixels
nx = 512
ny = 512
```

Figure 4

Configuration files for simulation with (a) an atomic and (b) a refractive index model. The source section defines the illumination properties, the particle section the sample properties and the detector section the parameters for the area pixel detector. Many parameters are optional and are set to default values if not specified. These configuration files together with the required structure files are included in the online repository of *Condor* and are located in the folder *examples_publication*.

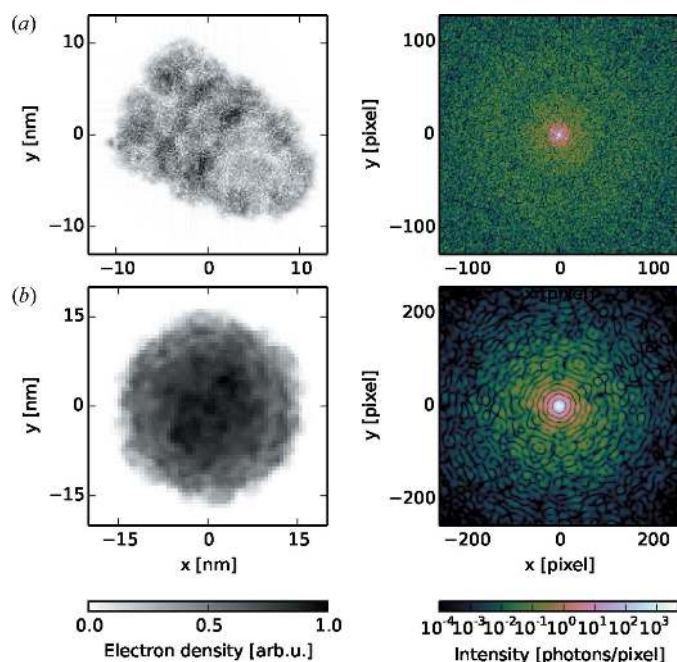


Figure 5

Simulation of diffraction patterns for two biological structures: (a) the GroEL–GroES complex and (b) the poliovirus particle. For each model the projection image is shown on the left and the noise-free intensity pattern on the right.

example configuration files, one for the calculation with an atomic model (Fig. 4a) and one for the calculation with a refractive index model (Fig. 4b). Every configuration file is subdivided into at least three sections [X-ray source, sample particle(s), pixel array detector]. All quantities follow the convention of the International System of Units. If a parameter is unspecified it is set to a default value. At the end of execution the results are written to an HDF5 file. The acronym HDF5 stands for Hierarchical Data Format version 5 (The HDF Group, 2016), which is a widely used file format for scientific applications and ensures high portability and performance. The data structure within the file follows the guidelines for the Coherent X-ray Imaging file format (Maia, 2012).

```
#!/usr/bin/env python
import condor
import numpy
from numpy.fft import fftn, fftshift

# X-ray source
photon = condor.utils.photon.Photon(energy_eV = 1500.)
source = condor.Source(wavelength = photon.get_wavelength(),
                      focus_diameter = 1000E-9,
                      pulse_energy = 1E-3)

# Sample particles
particles = {
  # Single proteins
  *particle_atoms_0":
    condor.ParticleAtoms(number = 0.6,
                        arrival = "random",
                        pdb_id = "1AON",
                        rotation_formalism = "random",
                        position_variation = "uniform",
                        position_spread = [50E-9, 50E-9, 1E-9]),

  # Water droplets
  *particle_spheroid_0":
    condor.ParticleSpheroid(number = 0.4, arrival = "random",
                           diameter = 8E-9,
                           diameter_variation = "normal",
                           diameter_spread = 2E-9,
                           flattening = 0.9,
                           flattening_variation = "uniform",
                           flattening_spread = 0.2,
                           material_type = "water",
                           position_variation = "uniform",
                           position_spread = [50E-9, 50E-9, 1E-9]),
}

# Area pixel detector (pnCCD)
detector = condor.Detector(distance = 0.15,
                          nx = 128-7,
                          ny = 128,
                          pixel_size = 600E-6,
                          x_gap_size_in_pixel = 7,
                          hole_diameter_in_pixel = 14,
                          noise = "poisson")

# Experiment object instance
experiment = condor.Experiment(source, particles, detector)

# Output file
W = condor.utils.cxiwriter.CXIWriter("./condor.cxi")

# Simulation of 5 images
for i in range(5):
  results = experiment.propagate()
  intensity_pattern = results["entry_1"]["data_1"]["data"]
  amplitudes_pattern = results["entry_1"]["data_1"]["data_fourier"]
  results["patterson_image"] = fftshift(fftn(fftshift(intensity_pattern)))
  results["projection_image"] = fftshift(fftn(fftshift(amplitudes_pattern)))
  W.write(results)
W.close()
```

Figure 6

Customized use of *Condor* by direct interaction with the Python API. The script simulates diffraction patterns from a mixture of GroEL–GroES complex and spheroidal water droplets of varying shape and size. A missing data region and Poisson noise are taken into account for the intensity estimate at the detector pixels. After the initialization, patterns are simulated sequentially in a loop by calling the method *propagate()* and appending the results to an HDF5 file.

The two example configuration files shown in Fig. 4 define experimentally feasible configurations at the LINAC Coherent Light Source (LCLS). The selected particle structures are the GroEL–GroES protein complex (Fig. 4*a*) and the poliovirus particle (Fig. 4*b*). The structure for the GroEL–GroES protein complex is taken from the atom positions of PDB entry 1aon (Xu *et al.*, 1997). The poliovirus particle is modelled by the density map derived from EMDB entry 1144 (Bubeck *et al.*, 2005). We projected the EMDB map to electron densities using experimentally determined values for atomic composition (Molla *et al.*, 1991) and mass density (Dans *et al.*, 1966) of poliovirus virions. Simulated results from these examples are shown in Fig. 5.

Condor provides not only intensities but also phases. Here the curvature of the Ewald sphere is small, and hence projection images in real space (left column in Fig. 5) can be readily calculated by inverse Fourier transforming the scattering amplitudes.

For a more customizable use, *Condor*'s application programming interface (API) can be called directly from any Python software. The *Condor* engine can thus be easily integrated into any software tool or pipeline that relies on simulated diffraction data. An example for a script that uses the *Condor* API is shown in Fig. 6. Projection images and diffraction patterns that were generated with this script are presented in Fig. 7. The script simulates an experiment where spheroidal water droplets contaminate the particle stream of GroEL–GroES protein complexes. Both particle species arrive in the scattering volume in random orientations and at random positions. The arrival statistics are modelled by a Poisson process with arrival rates of 0.2 for the water droplets and 0.9 for the protein complexes. The water droplets are not simulated as perfectly reproducible structures but as spheroids of varying size and shape. This is reflected in the model by size parameters that follow a normal distribution centred at 8 nm and values of the flattening parameter that follow a uniform distribution between 0.8 and 1.0.

4. Implementation

Condor is a Python package including C extensions for the computationally heavy operations. For the calculation of the

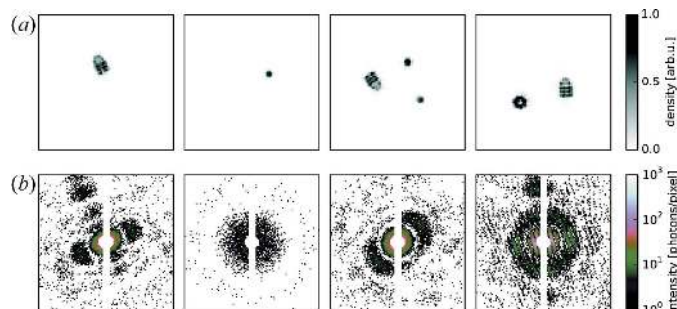


Figure 7 Simulation of diffraction patterns for a mixture of two particle species: the GroEL–GroES complex and spheroidal water droplets. (a) Real-space projection images and (b) respective simulated diffraction intensity patterns with Poisson noise and a pixel mask. The physical parameters resemble the conditions at the AMO beamline at the LCLS.

discrete Fourier transform in equations (6) and (11), *Condor* makes use of the non-equispaced fast Fourier transform (*NFFT*) C library (Keiner *et al.*, 2009). This library provides routines to calculate the discrete Fourier transform at non-equispaced points, for example on the curved surface of the Ewald sphere. For the refractive index model *Condor* deploys the common *NFFT* algorithm, which still requires equispaced sampling in the real-space domain. For the atomic model the generalized *NNFFT* algorithm is used, as it allows for non-equispaced sampling in both domains. The computation of the sums in the discrete Fourier transform can benefit from parallelization. Compilation with OpenMP (<http://openmp.org>) allows for an easy parallelization with moderate speed-ups. Diffraction from atomic models is normally more computationally demanding and here *Condor* supports the use of CUDA-capable graphics cards (<http://nvidia.com/cuda>), which can provide a drastic increase in performance.

Computation times were measured for the simulations of the examples shown in Figs. 4 and 5, which were carried out on a MacBookPro computer [2.5 GHz Intel Core i7 (4 cores, 8 threads), 16 GB 1600 MHz DDR3] equipped with a CUDA-capable graphics card (NVIDIA GeForce GT 750 M, 2048 MB memory). The atomic model included of 58 870 atom positions, and diffraction was predicted at 256×256 detector pixels. Using a single CPU and with CUDA disabled the calculation took 208 s. Enabling CUDA resulted in a computation time of 3 s, giving a speedup of $69.3\times$. The refractive index map consisted of $173 \times 173 \times 173$ voxels, and diffraction was predicted at 512×512 detector pixels. Using a single CPU the calculation took 19 s, and using four CPU threads it took 6.8 s, resulting in a speed-up of $2.8\times$.

Fig. 8(a) illustrates the representation of an experiment in *Condor* as a Python object. It contains a source object, one or

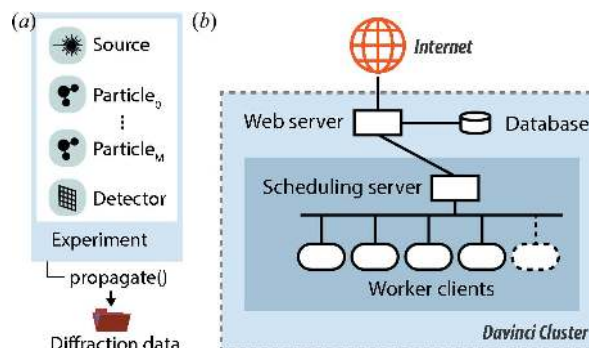


Figure 8 Implementation architecture. (a) An experiment is represented in *Condor* as a Python object that includes a source object, one or several sample particle objects, and a detector object. A call of the method `propagate()` starts a simulation and returns the diffraction data. (b) The web version of *Condor* is realized as a hierarchical client–server model. The web server provides a dynamic web page under the address <http://lmb.icm.uu.se/condor>. Under this page users can configure their experiment, and upon submission data are validated and then cached in a database. Simulation jobs are scheduled by a scheduling server that manages a network of worker clients. This worker farm is dynamically extended and shrunk depending on the number of requests. After completion of a simulation the web server presents previews and links to download to the user.

several particle objects, and a detector object. The experiment object has a method `propagate()` that starts the simulation of a single shot and returns the results in the form of a Python dictionary.

As an alternative to a local installation, *Condor* is also provided as a web application (Fig. 1) that supports most of the functionality of the full package. In the left panel of the web application one can configure the X-ray source, sample particle and detector. The upper right panel is used to submit simulation requests and monitor their progress. After a simulation has finished its results can be previewed and downloaded from the bottom right panel.

The web implementation of *Condor* is based on a Django (<https://www.djangoproject.com/>) web framework and uses a database for caching user inputs. The system is hosted by the Davinci GPU computer cluster of the Laboratory of Molecular Biophysics (Uppsala University, Sweden).

The architecture of the server–client model of the web implementation is illustrated in Fig. 8(b). When a user submits a simulation request the web server first checks the input. If the input passes validation the web server sends the requests to the *Condor* server, which manages a number of *Condor* clients. The first worker client that becomes available starts the *Condor* simulation. The number of worker clients is dynamically adjusted to the current load of the web page, such that at least one worker client is always available for processing a simulation request. The hierarchical architecture ensures responsiveness of the servers at all times, even when running multiple simulations simultaneously. While a simulation is running the scheduling server monitors the progress of the simulation. When finished the results are sent to the web server, which presents the user with previews and links for downloading the results as an HDF5 file.

5. Conclusion

FXI experiments at free-electron laser facilities are expensive and precious. Easy-to-use software can support researchers in improving data quality and can support data analysis. The software *Condor* is a fast simulation tool specialized for FXI research and covers a wide range of use cases and functionalities. Practically anybody is able to use *Condor* because of its simple structure and because common hurdles such as limited cross-platform compatibility or demanding hardware requirements have been avoided by making key features available through a web application. We, the developers, encourage and support the integration of the code into other software that relies on simulated FXI data. Reusability of the source code is facilitated by the availability of a simple and flexible Python API and by the distribution of the code under the Simplified BSD license.

Beyond its relevance in research *Condor* may be a useful educational software tool. Students may gain understanding of the laws of X-ray diffraction by studying changes in the diffraction pattern while changing experimental parameters. Moreover, entire experimental data sets can be readily simu-

lated by the students themselves. Students may be invited to pursue a reconstruction from simulated data.

In conclusion, *Condor* will enhance and stimulate collaborative activities in software development within the FXI community. Furthermore, the software will underpin efforts in FXI education, experiment planning, conducting of experiments, algorithm development and data validation.

References

- Als-Nielsen, J. & McMorrow, D. (2001). *Elements of Modern X-ray Physics*. New York: Wiley.
- Aquila, A. *et al.* (2015). *Struct. Dyn.* **2**, 041701.
- Ayyer, K., Geloni, G., Kocharyan, V., Saldin, E., Serkez, S., Yefanov, O. & Zagorodnov, I. (2015). *Struct. Dyn.* **2**, 041702.
- Bergh, M., Huldt, G., Timneanu, N., Maia, F. R. N. C. & Hajdu, J. (2008). *Q. Rev. Biophys.* **41**, 181–204.
- Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. (2000). *Nucleic Acids Res.* **28**, 235–242.
- Bogan, M. J. *et al.* (2008). *Nano Lett.* **8**, 310–316.
- Brown, P. J., Fox, A. G., Maslen, E. N., O’Keefe, M. A. & Willis, B. T. M. (2006). *International Tables for Crystallography*, Vol. C, 1st online ed., edited by E. Prince, ch. 6.1. Chester: International Union of Crystallography.
- Bubeck D., Filman, D. J., Cheng, N., Steven, A. C., Hogle, J. M. & Belnap, D. M. (2005). *J. Virol.* **79**, 7745–7755.
- Chapman, H. N., Barty, A. *et al.* (2006). *Nat. Phys.* **2**, 839–843.
- Chapman, H. N. & Nugent, K. A. (2010). *Nat. Photon.* **4**, 833–839.
- Dans, P. E., Forsyth, B. R. & Chanock, R. M. (1966). *J. Bacteriol.* **91**, 1605–1611.
- Daurer, B. J., Hantke, M. F., Nettelblad, C. & Maia, F. R. N. C. (2016). *J. Appl. Cryst.* **49**, 1–6.
- Ekeberg, T. *et al.* (2015). *Phys. Rev. Lett.* **114**, 098102, 1–6.
- Feigin, L. A. & Svergun, D. I. (1987). *Structure Analysis by Small-Angle X-ray and Neutron Scattering*. New York: Plenum Press.
- Hamzeh, F. M. & Bragg, R. H. (1974). *J. Appl. Phys.* **45**, 3189–3195.
- Hantke, M. F. *et al.* (2014). *Nat. Photon.* **8**, 943–949.
- Henke, B. L., Gullikson, E. & Davis, J. (1993). *At. Data Nucl. Data Tables*, **54**, 181–342.
- Keiner, J., Kunis, S. & Potts, D. (2009). *ACM Trans. Math. Softw.* **36**, 1–30.
- Lawson, C. L. *et al.* (2011). *Nucleic Acids Res.* **39**, D456–D464.
- Loh, N. D. *et al.* (2012). *Proc. SPIE*, **8504**, 850403.
- Maia, F. R. N. C. (2012). *Nat. Methods*, **9**, 854–855.
- Molla, A., Paul, A. & Wimmer, E. (1991). *Science*, **254**, 1647–1651.
- Neutze, R., Wouts, R., van der Spoel, D., Weckert, E. & Hajdu, J. (2000). *Nature*, **406**, 752–757.
- Paganin, D. M. (2006). *Coherent X-ray Optics*. Oxford University Press.
- Schot, G. van der *et al.* (2015). *Nat. Commun.* **6**, 5704.
- Seibert, M. M. *et al.* (2011). *Nature*, **470**, 78–81.
- Serkez, S., Kocharyan, V. & Saldin, E. (2013). *35th International Free-Electron Laser Conference*, pp. 574–582. Red Hook: Curran Associates.
- The HDF Group (2016). Hierarchical Data Format. Version 5. <http://www.hdfgroup.org/HDF5/>.
- Winn, M. D. *et al.* (2011). *Acta Cryst.* **D67**, 235–242.
- Xu, Z., Horwich, A. L. & Sigler, P. B. (1997). *Nature*, **388**, 741–750.
- Yefanov, O. M. & Vartanyants, I. A. (2013). *J. Phys. B At. Mol. Opt. Phys.* **46**, 164013.
- Yoon, C. H., Yurkov, M. V., Schneidmiller, E. A., Samoylova, L., Buzmakov, A., Jurek, Z., Santra, R., Loh, N. D. & Mancuso, A. P. (2015). *Sci. Rep.* **6**, 24791.