Outline of Data Processing Steps for CDI/FCDI: Ver 2.1

Note that italics indicates tasks specific to FCDI

Acronyms: Entrance Slits (ENS), Exit Slits (EXS), Region of Interest (ROI), Signal to Noise Ratio (SNR), White Field (WF), Dark Field (DF) , Numerical Aperture (NA), Error Reduction (ER), Hybrid Input-Output (HIO) , Exit Surface Wave (ESW).

0) Confirm all experimental parameters and geometry

Key quantities: detector pixel size, wavelength, *focal length*, sample-detector distance, *focus-sample distance*¸ coherence properties of beam (e.g. ENS/EXS), exposure time.

Note that these quantities can be found in the lab book (electronically stored on Osiris). To determine the focal length you need to know the radius of the zone plate which is 80 microns.

1) View HDF files

Common tasks:

- ***look at data on a log scale***

-***2 x 2 and 4 x 4 binning to cycle through images:*** (input: individual data, WF and DF files, output: GUI image) all the images in a stack, normally 100-600 are queued and the resolution of each reduced by 2 x 2 or 4 x 4 binning so that they can be viewed quickly in a GUI. Normally we can scroll back and forth through images using the arrow keys. Program used: Ifran view. Priority: Extension work

-***plot average intensity Vs frame number, full-frame and ROI:*** (input: individual data, WF, DF files, output: GUI line plot) produce a line plot of the mean intensity for the entire frame or a small area of the frame (defined by an x,y starting point and the dimensions of the rectangle- it would be useful to see the ROI in a GUI) as a function of the frame number. Program used: MATLAB. Priority: Very Useful.

-***check for saturation/dead pixels***: (input: individual data, WF, DF files, output: GUI image) checking for pixels which are dead (zero counts even when neighbouring pixel values are high), looking for pixels which are 'hot' (max counts always), or which are otherwise damaged (significantly different response to neighbouring pixels). Normally in a GUI use a sliding bar to highlight pixels above or below a certain threshold. With those threshold values fixed we cycle through the images. Program used: Image J or MATLAB. Priority: Very Useful.

-***run correlations on data and WFs***: (input: individual data, WF, DF files, output: GUI line plot) calculate the cross-correlation coefficient of each subsequent frame with a specified frame in the stack. Plot the result as cross-correlation vs. frame number. Program used: C (calccrossR) or MATLAB. Priority: Very Useful.

2) Convert HDF files and extract header info

Common tasks:

-***convert HDF to bin, Tif, ppm or Gif (not for processing)*** (input: individual data, WF, DF files, output: file, no GUI) . Program used: c (HDF read) or MATLAB. Priority: Essential

-***Extract ring current, motor positions etc. from HDF*** (input: individual data, WF, DF files, output: parameter file). Program used: MATLAB. Priority: Very Useful

3) Raw data processing

Common tasks:

-darkfield subtraction, may need to rescale DF to data/WF to optimise SNR: (input: individual data, WF, DF files, output: GUI image check and modified file) just multiplying the DF by a constant before subtracting it from the WF to remove noise (especially important at high angle) whilst preserving the scattered signal. Result should be absolute, no negative numbers. Usually need to view the data on a log scale as the ratio of DF to WF is adjusted. Also may need to subtract a pedestal- i.e. a threshold below which everything is set to zero. Program used: c (subtract2sp4) or MATLAB. Priority: Essential

-check beam stability in data and WF: (input: individual data, WF files, output: GUI line plot) usually we monitor differences in the beam intensity over time (more difficult when a sample is in place). Check the cross-correlation coefficients for the WF and data. and measure the positions of fringes around the illumination as a function of frame number. Program used: c (calccrossR) or MATLAB. Priority: Very Useful

-check correlations again after DF subtraction: (input: individual data, WF, DF files, output: GUI line plot) having subtracted a correctly scaled DF (see above) we re-calculate the cross-correlation coefficients, as a constant background noise (e.g. light leaking into the chamber), will give an artificially high correlation value. Program used: c (calccrossR) or MATLAB. Priority: Very Useful

-sum frames, check NA of detected scatter: (input: individual processed data files, output: GUI line plot, summed image) after DF subtraction, frames which are well correlated, and for which the beam/sample position is stable are added together. The result is viewed on a log scale to determine what the NA of the detected scatter is. This is a bit subjective, but I usually look for the last 'complete' fringes on the detector. Program used: c (2dplot) or MATLAB. Priority: Very Useful

-check normalisation of WF and data using ROI: (input: summed data, WF files, output: GUI line plots and ROI image) In the detector plane the subtraction of the WF in areas where there is no sample should be zero (alternatively their ratio should be 1). If the sample fills the entire illumination, the normalisation may need to be optimised as part of the reconstruction. Program used: c (sp4divide) or MATLAB. Priority: Very Useful

-look at autocorrelation: (input: summed data, output: GUI image) take the FT of the intensity data to obtain the autocorrelation function for the diffracting object. The initial object support should occupy no more than 50% of the area of the autocorrelation function. Program used: c (dofft and 2dplot) or MATLAB. Priority: Very Useful

4) Pre-reconstruction

Common tasks:

-construct simulated test object and propagate to far-field to sanity check data and experimental parameters: (input: simulated data, output: GUI image) draw a shape which approximately corresponds to the test object with a pixel value of 1 wherever within the object. Assuming that the object is uniform thickness and refractive index, create two arrays, one where the object array is multiplied by the imaginary part of the refractive index for the object material and another where the object array is multiplied by the phase of the object material. The two arrays form a complex array describing the amplitude and phase of the diffracting object according to Aexp(iPhi), where A is the amplitude and Phi is the phase. Applying the correct curvature and transforming gives the sample ESW at the detector (for plane wave just taking the FT gives the Fraunhofer diffraction pattern). Program used: c (GIMP, sp4multiply, addphases) or MATLAB. Priority: Essential

-Form an initial support based on e.g. autocorrelation, optical microscopy, beam size: (input: a priori data, output: GUI image, support array) support is just larger than the dimensions of the sample ESW and is binary. Program used: c (GIMP) or MATLAB. Priority: Extension Work

-Reconstruct the phase of the WF using 3-plane propagation: (input: WF data, output: GUI image, reconstructed WF). Program used: c (holoscatterfit) or MATLAB. Priority: Essential

-carry out holographic 1 step reconstruction : (input: WF and data frames, output: GUI image, reconstructed WF) mask out the high angle scatter and propagate ESW back to sample plane to view a holographic image of the sample. Program used: c (subtract2sp4, dofft) or MATLAB. Priority: Essential

-Carry out reconstruction, usually a combination of HIO and ER, once solution looks like it is close to converging just use ER. (input: summed data, output: GUI image). Program used: c (holoscatterfit2) or MATLAB. Priority: Essential

-Periodically, if necessary, update support manually or use SW (Gauss convolution) with parameter update. (input: reconstructed data, output: GUI image). Program used: c (holoscatterfit2), GIMP or MATLAB. Priority: Essential

-refine on the experimental parameters (normally focus-sample) to give the 'sharpest' edge features (input: reconstructed data, output: GUI line plot) Program used: c (2dcrop, 2dplot, sp42ASCII) or MATLAB. Priority: Essential

5) Post reconstruction

-look at sample ESW, are there signs of stagnation? (i.e. often get stuck between 2 mirror solutions) (input: reconstructed ESW, output: GUI image) : can usually see two solutions for the object but mirrored w.r.t. one another overlapping. Program used: c (2dplot) or MATLAB. Priority: Extension Work

-look at transmission function, are amplitude and phase constant with object properties? is there any phase wrapping?: (input: reconstructed ESW, output: GUI line plot) Are there sudden and discrete jumps in the phase where the sign suddenly changes, such that multiplying by -1 would bring the phase back to make a smooth continuous function. Program used: c (2dplot) or MATLAB. Priority: Extension Work

-compare ESW at detector to measured data 'by eye' are the two consistent? are there any 'artefacts' which are obvious: (input: reconstructed ESW and data, output: GUI image) look for strange 'blobs' etc in the diffraction not consistent with what you would expect for the sample ESW. Program used: c (2dplot) or MATLAB. Priority: Extension Work

-estimate resolution- take lineout through sharp features in object. Look at the outermost reconstructed fringes in detector ESW. (input: reconstructed ESW, output: GUI line plot and data).

Program used: c (2dplot and 2dcrop) or MATLAB. Priority: Extension Work