JOURNAL OF APPLIED CRYSTALLOGRAPHY

Volume 54 (2021)

Supporting information for article:

Geometrical-optics formalism to model contrast in dark-field X-ray microscopy

H. F. Poulsen, L. E. Dresselhaus-Marais, M. A. Carlsen, C. Detlefs and G. Winther

Supplementary material

Geometrical Optics Formalism to Model Contrast in Dark-Field X-ray Microscopy

H. F. Poulsen, L. E. Dresselhaus-Marais, M. A. Carlsen, C. Detlefs, G. Winther

This Supplementary Material comprises MATLAB codes which are referred to in main text

1) recspace_res.m that generates a model for the reciprocal space resolution function
2) forward.m that generates a forward model for one image
3) image_range.m is an auxiliary plotting function

1. Reciprocal space resolution function simulator

```
function [Resq_i ratio_outside] = recspace_res(Nrays, qi1_range, npoints1,
qi2_range, npoints2, qi3_range, npoints3, plot_figs);

% Model for rec. space resolution function for DFXM forward simulation paper.
% The objective is modelled as an isotropic Gaussian with an NA and in addition
% a square phyical aperture of d¬¥side length D.

% H.F. Poulsen, June 16, 2020, version 1.0
% H.F. Poulsen, Jan 22, 2021, version 1.1

% input parameters:
%   Nrays: numer of rays to be used
%   qi1_range, qi2_range, qi3_range: ranges for Resq_i in crystal system, in
inverse AA.
%   npoints1,  npoints2, npoints3:  nr of points within each range
%   plot_figs: a flag; if 1 then plots will be generated, otherwise not
% output parameters:
%   Resq_i: voxelized rec. space resolution function in the IMAGING system.
%                                       Normalised to a max value of 1
%   ratio_outside: The fraction of rays not within the range defined by
%                                       qi1_range, qi2_range, qi3_range.


%Instrumental parameters
zeta_v_fwhm = 0.53E-3; % incoming divergence in vertical direction, in rad
zeta_h_fwhm = 1E-5; % incoming divergence in horizontal direction, in rad
NA_rms = 7.31E-4/2.35; % NA of objective, in rad
eps_rms = 0.00006; % rms width of x-ray energy bandwidth
theta = 20.73/2*pi/180; % scattering angle, in rad
D = 2*sqrt(50E-6*1E-3); % physical aperture of objective, in m
d1 = 0.274; % sample-objective distance, in m
phys_aper = D/d1;


%%%%  Ray racing in crystal system  %%%%
% These are Equations 43-45 in DFXM paper

for ii=1:Nrays

    %Define the properties of one ray
    zeta_v(ii) = (rand()-0.5)*zeta_v_fwhm; %top-hat
    zeta_h(ii) = randn()*zeta_h_fwhm/2.35;  %Gaussian
    eps(ii) = randn()*eps_rms;

    for kk=1:100 %find a trial delta_2theta
        test = randn()*NA_rms;
        if abs(test) < (phys_aper/2)
            delta_2theta(ii) = test;
            break
        end
    end
    for kk=1:100 %find a trial xi (equal to cos(theta)*eta )
        test = randn()*NA_rms;
        if abs(test) < (phys_aper/2)
            xi(ii) = test;
            break
        end
    end

    % Convert to crystal system cooridnates
    qrock(ii) = -zeta_v(ii)/2 - delta_2theta(ii)/2;
    qroll(ii) = -zeta_h(ii)/(2*sin(theta)) -  xi(ii)/(2*sin(theta));
    qpar(ii) = eps(ii) + cot(theta)*(-zeta_v(ii)/2 + delta_2theta(ii)/2);
```

```
end

% Convert from crystal to imaging system  (Eq. 41 in DFXM paper)
qrock_prime = cos(theta)*qrock + sin(theta)*qpar;
q2th =  - sin(theta)*qrock + cos(theta)*qpar;

% Convert point cloud into local density function, Resq_i, normalised to 1
% If the range is set too narrow such that some points falls outside ranges,
%             the fraction of points outside is returned as ratio_outside
Resq_i = zeros(npoints1, npoints2, npoints3);
outside = 0;
for ii = 1:Nrays
    index1 = floor( (qrock_prime(ii)+ qi1_range/2)/qi1_range*(npoints1-1) ) + 1;
    index2 = floor( (qroll(ii)+ qi2_range/2)/qi2_range*(npoints2-1) ) + 1;
    index3 = floor( (q2th(ii) + qi3_range/2)/qi3_range*(npoints3-1) ) + 1;
    if (index1 < 1) | (index1 > npoints1) | (index2 < 1) | (index2 > npoints2) |
(index3 < 1) | (index3 > npoints3)
        outside = outside + 1; continue
    else
        Resq_i(index1,index2, index3) = Resq_i(index1,index2, index3) + 1;
    end
end
ratio_outside = outside/Nrays;
Resq_max = max(max(max(Resq_i)));
Resq_i = Resq_i/Resq_max; %normalise to 1 as max value

%%%%%%%%%%%%  For test purposes, plots %%%%%%%%%%%

if plot_figs == 1

    % Scatter plot  with shadows
    % not feasible/relevant  if Nrays > 1 million
    % a poltting range is required (same in all directions)

    plot_half_range = 0.0045;
    if Nrays < 1000001
        %in crystal system
        figure;    scatter3(qrock,qroll,qpar,'SizeData',1);
        xlabel('$\hat{q}_{rock}$','Interpreter','latex');
ylabel('$\hat{q}_{roll}$','Interpreter','latex');
zlabel('$\hat{q}_{par}$','Interpreter','latex');
        set(gca,'XLim',[-plot_half_range plot_half_range],'YLim',[-plot_half_range
plot_half_range],'ZLim',[-plot_half_range plot_half_range]);
        set(gca,'FontSize',15);
        hold on
        plot3(zeros(size(qrock))+plot_half_range, qroll, qpar,'o','MarkerSize',1);
        plot3(qrock, zeros(size(qroll))+plot_half_range, qpar,'o','MarkerSize',1);
        plot3(qrock, qroll, zeros(size(qpar))-plot_half_range,'o','MarkerSize',1);
        hold off

        % in image system
        figure; scatter3(qrock_prime,qroll,q2th,'SizeData',1);
        xlabel('$\hat{q}^{\prime}_{rock}$','Interpreter','latex');
ylabel('$\hat{q}_{roll}$','Interpreter','latex');
zlabel('$\hat{q}_{2\theta}$','Interpreter','latex');
        set(gca,'XLim',[-plot_half_range plot_half_range],'YLim',[-plot_half_range
plot_half_range],'ZLim',[-plot_half_range plot_half_range]);
        set(gca,'FontSize',15);
        hold on
        plot3(zeros(size(qrock_prime))+plot_half_range, qroll,
q2th,'o','MarkerSize',1);
        plot3(qrock_prime, zeros(size(qroll))+plot_half_range,
q2th,'o','MarkerSize',1);
        plot3(qrock_prime, qroll, zeros(size(q2th))-
plot_half_range,'o','MarkerSize',1);
        hold off

    end
```

```
    % Central slices of Resq_i along three main directions
    % Note that images are rotated by 90 deg to comply with scatterplot
    %       conventions (related to MATLABs plotting conventions)
    slicenr = round(npoints1/2);
    im1 = squeeze(Resq_i(slicenr,:,:));
    qi2_start = -qi2_range/2; qi2_step = qi2_range/npoints2;
    X = [qi2_start: qi2_step: qi2_start + (npoints2-1)*qi2_step]; % create x-axis
    qi3_start = -qi3_range/2; qi3_step = qi3_range/npoints2;
    Y = [qi3_start: qi3_step: qi3_start + (npoints3-1)*qi3_step]; % create y-axis
    figure; imagesc(X,Y,rot90(im1),[0,1]);
    xlabel('$q_{roll}$','Interpreter','latex');
    ylabel('$q_{2\theta}$','Interpreter','latex');
    colorbar; title('Slice through center in $q_{rock_prime}$ ','FontSize',11,
'Interpreter','latex');

    slicenr = round(npoints2/2);
    im2 = squeeze(Resq_i(:,slicenr,:));
    qi1_start = -qi1_range/2; qi1_step = qi1_range/npoints1;
    X = [qi1_start: qi1_step: qi1_start + (npoints1-1)*qi1_step]; % create x-axis
    qi3_start = -qi3_range/2; qi3_step = qi3_range/npoints3;
    Y = [qi3_start: qi3_step: qi3_start + (npoints3-1)*qi3_step]; % create y-axis
    figure; imagesc(X,Y,rot90(im2),[0,1]);
    xlabel('$q_{rock,prime}$','Interpreter','latex');
    ylabel('$q_{2\theta}$','Interpreter','latex');
    colorbar; title('Slice through center in $q_{roll}$ ','FontSize',11,
'Interpreter','latex');

    slicenr = round(npoints3/2);
    im3 = squeeze(Resq_i(:,:,slicenr));
    qi1_start = -qi1_range/2; qi1_step = qi1_range/npoints1;
    X = [qi1_start: qi1_step: qi1_start + (npoints1-1)*qi1_step]; % create x-axis
    qi2_start = -qi2_range/2; qi2_step = qi2_range/npoints2;
    Y = [qi2_start: qi2_step: qi2_start + (npoints2-1)*qi2_step]; % create y-axis
    figure; imagesc(X,Y,rot90(im3),[0,1]);
    xlabel('$q_{rock,prime}$','Interpreter','latex');
    ylabel('$q_{roll}$','Interpreter','latex');
    colorbar; title('Slice through center in $q_{2\theta}$ ','FontSize',11,
'Interpreter','latex');

    % Projection of Resq_i along one of the main  with correct aspect ratio,
    % please add others if needed
    % Note that images are rotated by 90 deg to comply with scatterplot
    %         conventions (related to MATLABs plotting conventions)
    project = zeros(npoints1,npoints3);
    for ii=1:npoints1
        for jj= 1: npoints3
            project(ii,jj) = sum(Resq_i(ii,:,jj));
        end
    end
    qi1_start = -qi1_range/2; qi1_step = qi1_range/npoints1;
    X = [qi1_start: qi1_step: qi1_start + (npoints1-1)*qi1_step]; % create x-axis
    qi3_start = -qi3_range/2; qi3_step = qi3_range/npoints3;
    Y = [qi3_start: qi3_step: qi3_start + (npoints3-1)*qi3_step]; % create y-axis
    figure; imagesc(X,Y,rot90(project),[0,1]);
    colorbar;
    title('Title','FontSize',11)
    xlabel('$q_{rock,p}','Interpreter','latex');
    ylabel('$q_{2\theta}','Interpreter','latex');
    colorbar; title('Projection $q_{roll}$ ','FontSize',11, 'Interpreter','latex');
    daspect([1,1,1]);
end

end
```

2. Forward model simulator for one image

```
function [im, qi_field] = forward(phi,chi,TwoDeltaTheta,U, h, k, l,plot_im,
max_int, plot_qi)
% Forward Model for DFXM
% It generates one image, im, using eq 58 in the paper. It operates in teh
imaging
% coordinate systen and assumes a simplified geometry with input of phi,
chi
% and 2DeltaTheta. For now it also assumed thas the crystal has cubic
symmetry,
% that omega=0 and mu = theta_0.  The density is set
% to 1 everywhere. The detector plane normal is along the
% optical axis of the objective.These restrictions are easuly overcome.

% The  reciprocal space resolution function is  loaded in as a MATLAB m
% file and the q-ranges input below.
% The displacement gradient function, Fg, is read in and assumed to be
expressed
% in the sample coordinate system.
% Relevant experimental paramteres are set internally in this function.


% It calls the function providind Fg. PLEASE INSERT THE RIGHT NAME FOR THIS
% FUNCTION. Currently  Fg = Fg_test(r_g)

% input parameters:
%   phi: rotation in phi in rad
%   chi: rotation in phi in rad
%   TwoDeltaTheta:  offset in 2theta in rad
%   U:  3x3 matrix with teh orientation of teh grain
%   h, k, l:  Miller indices
%   plot_im: a flag; if 1 then a plot of the image im will be generated,
otherwise not
%   max_int: defines max intensity in plots. If max_int=0 then the images
%            are scaled automatically
%   plot_qi: a flag; if 1 then plots of qi will be generated, otherwise
not. Mainly for debugging
% output parameters:
%   im: simulated image without magnification.
%   NN: dimension of qi_field,fordebugging
%   qi_field: output of qi(x,y,z) for debugging. Note not same steps.


% H.F. Poulsen, June 16, 2020, version 1.0
% H.F. Poulsen, Jan 31, 2021, version 2.0


%%%%%%%%%%       INPUT       %%%%%%%%%%%%%%%

%INPUT FOV
Npixels = 50; % nr of pixels on detector (same in both y and z) - sets the
FOV.
% even, such that 0 is between two pixels
Nsub = 2;  % NN^3 = (Nsub*Npixels)^3 is the total number of "rays" probed
NN = Nsub*Npixels;
```

```
%INPUT instrumental settings, related to direct space resolution function
psize = 75E-9; %pixel size in units of m, in the object plane
zl_rms = 0.6E-6/2.35  %rms value of Gaussian beam profile, in m, centered
at 0
theta_0 = 20.73/2*pi/180; %in rad
%  input reciprocal space resolution function (in the imaging system)
%  by loading an already generated version Resq_i and inserting q_i-ranges
and steps here
load('Resq_i.mat','Resq_i');
qi1_range = 8E-3;  npoints1 = 40;
qi2_range = 8E-3; npoints2 = 40;
qi3_range = 8E-3; npoints3 = 40;


%%%%%%%%%%       DETERMINE IMAGE, Im      %%%%%%%%%%%%%%%

theta = theta_0; %+ TwoDeltaTheta;
yl_start = -psize*Npixels/2 + psize/(2*Nsub); %start in yl direction, in
unit of m, centered at 0
yl_step = psize/Nsub;
xl_start = ( -psize*Npixels/2 + psize/(2*Nsub) )/tan(2*theta); % start in
xl direction, in m, for zl=0
xl_step = psize/Nsub/tan(2*theta);
zl_start = -0.5*zl_rms*6; % start in zl direction, in m, for zl=0
zl_step = zl_rms*6/(NN-1);


qi1_start = -qi1_range/2; qi1_step = qi1_range/(npoints1-1);
qi2_start = -qi2_range/2; qi2_step = qi2_range/(npoints2-1);
qi3_start = -qi3_range/2; qi3_step = qi3_range/(npoints3-1);


Q_norm = sqrt(h^2 + k^2 +l^2); % We have assumed B_0 = I
q_hkl = [h k l]'/Q_norm;


mu = theta_0; M =  [cos(mu) 0 sin(mu); 0 1 0; -sin(mu) 0 cos(mu)];
Omega = eye(3);
Chi = eye(3);
Phi = eye(3);
Gamma = M*Omega*Chi*Phi;
Theta = [cos(theta) 0 sin(theta); 0 1 0; -sin(theta) 0 cos(theta)];

im = zeros(Npixels,Npixels);  %The forward model image
qi_field = zeros(NN,NN,NN,3);

% The tensor fields Fg, Hg and the vector fields qs,qc, qi are all defined
% as 3D fields in the lab system
for ii=1:NN
    yl = yl_start +(ii-1)*yl_step; % uniform beam in yl

    for jj=1:NN
        zl = zl_start +(jj-1)*zl_step;
        prob_z = exp(-0.5*(zl/zl_rms)^2);

        for kk=1:NN
            xl = xl_start +(kk-1)*xl_step + zl/tan(2*theta);
```

```
            rl = [xl yl zl]';

            det_index_y = floor((ii-1)/Nsub) + 1; % THIS ALIGNS WITH yl
            det_index_z = floor((kk-1)/Nsub) + 1; % THIS IS THE OTHER
DETECTOR DIRECTION AND FOLLOWS xl BUT WITH MAGNIFICATION

            % Determine Fg for given voxel
            rs = Gamma'*rl; %NB: Gamma inverse Eq. 5
            rg = U'*rs; %NB U inverse, Eq. 7
            xg = rg(1);  yg = rs(2); zg = rs(3);
            Fg(1:3,1:3)  = Fg_test(xg*1E6,yg*1E6,zg*1E6); %multiply by 1E6
as Fg_test is expressed in microns

            %  Determine qi for given voxel
            Hg = inv(Fg)'- eye(3); % Eq. 31
            qs = U*Hg*q_hkl;   % Eq 32
            qc = qs + [phi - TwoDeltaTheta/2; chi;
(TwoDeltaTheta/2)/tan(theta_0)]; % Eq 40 (also Eq. 20)
            qi = Theta*qc; % Eq 41
            qi_field(kk,ii,jj,1:3) = qi; % for plotting, sorted in order
x_l,y_l,z_l,1:3

            % Interpolation in rec. space resolution function.
            index1 = floor( (qi(1) - qi1_start)/qi1_step ) + 1;
            index2 = floor( (qi(2) - qi2_start)/qi2_step  ) + 1;
            index3 = floor( (qi(3) - qi3_start)/qi3_step  ) + 1;

            % Determine intensity contribution from voxel based on
rec.space res.function
            if index1 > 0 & index2 > 0 & index3 > 0 & index1 < npoints1+1 &
index2 < npoints2+1 & index3 < npoints3+1
                prob = Resq_i(index1,index2,index3)*prob_z;
            else
                prob = 0;
            end
            im(det_index_z, det_index_y) = im(det_index_z, det_index_y) +
prob;

        end
    end
end


%%%%%%%%%%       PLOT THE IMAGE, Im      %%%%%%%%%%%%%%%


if (plot_im ==1)
    Y = [xl_start: xl_step: xl_start + (NN-1)*xl_step]*1E6; % create
vertical axis in plot
    X = [yl_start: yl_step: yl_start + (NN-1)*yl_step]*1E6; % create
horizoncal axis in plot

    figure;
    if max_int==0
        image_range(X,Y,im,0,0,max_int); %plot with scale set by MATLAB
    else
        image_range(X,Y,im,1,0,max_int); %plot with range [0 max_int]
    end
    set(gca,'ydir','normal')
```

```
    ylabel('$x_{\ell}$ ($\mu$m)','Interpreter','latex'); xlabel('$y_{i}$
($\mu$m)','Interpreter','latex');
    colorbar;
    title(['Fw proj: $\phi =$ ', num2str(phi), '  \\ , $\chi =$',
num2str(chi),  ', $\Delta 2\theta =$',
num2str(TwoDeltaTheta)],'FontSize',11, 'Interpreter','latex')
    daspect([1 ,1, 1]);
end

%%%%%%%%%%       PLOT PROJECTIONS OF q_i      %%%%%%%%%%%%%%%
% THe field is in direct space expressed in the lab system.

if (plot_qi == 1)
    X = [xl_start: xl_step: xl_start + (NN-1)*xl_step]*1E6; % rulers on
axis in mu
    Y = [yl_start: yl_step: yl_start + (NN-1)*yl_step]*1E6; % rulers on
axis in mu
    Z = [zl_start: zl_step: zl_start + (NN-1)*zl_step]*1E6; % rulers on
axis in mu

    %Plot of three components of qi in (x,y, z=0) plane
    figure;
    subplot(1,3,1); %plot qi_1
    image_range(Y,X,squeeze(qi_field(:,:,floor(NN/2),1)),1,-1E-4,1E-4);
    set(gca,'ydir','normal');
    colorbar;
    title('qi_1 for (x,y) plane, z=0');
    ylabel('$x_{\ell}$ ($\mu$m)','Interpreter','latex'); xlabel('$y_{\ell}$
($\mu$m)','Interpreter','latex');
    daspect([1 ,1, 1]);

    subplot(1,3,2); %plot qi_2
    image_range(Y,X,squeeze(qi_field(:,:,floor(NN/2),2)),1,-1E-4,1E-4);
    set(gca,'ydir','normal');
    colorbar;
    title('qi_2 for (x,y) plane, z=0');
    ylabel('$x_{\ell}$ ($\mu$m)','Interpreter','latex'); xlabel('$y_{\ell}$
($\mu$m)','Interpreter','latex');
    daspect([1 ,1, 1]);

    subplot(1,3,3); %plot qi_3
    image_range(Y,X,squeeze(qi_field(:,:,floor(NN/2),3)),1,-1E-4,1E-4);
    set(gca,'ydir','normal');
    colorbar;
    ylabel('$x_{\ell}$ ($\mu$m)','Interpreter','latex'); xlabel('$y_{\ell}$
($\mu$m)','Interpreter','latex');
    title('qi_3 for (x,y) plane, z=0');
    daspect([1 ,1, 1]);

    %Plot of three components of qi in (x,y=0, z) plane
    figure;
    subplot(1,3,1); %plot qi_1
    image_range(Z,X,squeeze(qi_field(:,floor(NN/2),:,1)),-1E-4,1,1E-4);
    set(gca,'ydir','normal');
    colorbar;
    title('qi_1 for (x,z) plane, y=0');
    ylabel('$x_{\ell}$ ($\mu$m)','Interpreter','latex'); xlabel('$z_{\ell}$
($\mu$m)','Interpreter','latex');
    daspect([1 ,1, 1]);
```

```
    subplot(1,3,2); %plot qi_2
    image_range(Z,X,squeeze(qi_field(:,floor(NN/2),:,2)),1,-1E-4,1E-4);
    set(gca,'ydir','normal');
    colorbar;
    title('qi_2 for (x,z) plane, y=0');
    ylabel('$x_{\ell}$ ($\mu$m)','Interpreter','latex'); xlabel('$z_{\ell}$
($\mu$m)','Interpreter','latex');
    daspect([1 ,1, 1]);

    subplot(1,3,3); %plot qi_3
    image_range(Z,X,squeeze(qi_field(:,floor(NN/2),:,3)),1,-1E-4,1E-4);
    set(gca,'ydir','normal');
    colorbar;
    ylabel('$x_{\ell}$ ($\mu$m)','Interpreter','latex'); xlabel('$z_{\ell}$
($\mu$m)','Interpreter','latex');
    title('qi_3 for (x,z) plane, y=0');
    daspect([1 ,1, 1]);

end

end
```

3. Plotting function image_range.m

```
function [im_out]=  image_range(X,Y,im_in, flag, lower, upper)

% This makes an image using imagesc but with the added function that if
% flag is set to 1 then the intensity range is fixed between lower and
upper.

% Ver 1.0 H.F. Poulsen Jan 31, 2021

if flag ==1
    im_temp = max(lower,im_in);
    im_temp = min(upper, im_temp);
    im_temp(1,1) = lower; %hack to ensure there is one point with value of
lower
    im_temp(1,2) = upper; %hack to ensure there is one point with value of
upper
    im_out =im_temp;
else
    im_out =im_in;
end
imagesc(X,Y,im_out);

end
```