

SUPPLEMENTARY MATERIAL

For “Simulation of Small-Angle Scattering (SAXS or SANS) Curves by Numerical Fourier Transformation”

(Klaus Schmidt-Rohr, Iowa State University, srohr@iastate.edu)

MatLab code of the algorithm depicted in Figure 1(a):

```
% SAXS I(q) (orientationally averaged cross section)
% from 3D density rho(x)

clear lq
clear lq3D
clear lscatt
clear sincsqr

nx=input('# of pts in each dimension (e.g. 256)-->');
nxf=input('# of q points in I(q) (e.g. 128 to be faster) -->');
% can be smaller than nx, to speed up calculation
% (gives only small-q results)

nxd2=nx/2;
nxfd2=nxf/2;
iqcent=nxd2+1;

%-----
% From outside: scattering density rho
% make sure nx is the same in the external file
%-----
```

```

% -----
% Calculate scattering amplitude A(q) (complex)
% by Fourier transformation of rho(x)
% I(q)=|A(q)|^2
% -----

Iq3D=abs(fftshift(rho)).^2;
Iq3D=fftshift(Iq3D); %center of grid (nxd2+1) is at q=0

%-----
% Prepare convolution with elementary square of area a^2
% by multiplication with sinc function in each dimension
%-----

iq1=(1:nx)-iqcent;
q1ad2=pi*iq1/nx+0.00000001; % q1*a/2 = iq1c*2pi/(nx*a) *a/2
sincsqr=(sin(q1ad2)./q1ad2).^2;

%-----
% Now calculate I(|q|)*q^2 by summing over all orientations
% this is I(q)*q^2, because the density of the cubic grid varies like q^2
%-----

Iscatt=zeros(nxf,1);

iqmin=iqcent-nxfd2;
iqmax=iqcent+nxfd2-1;
symweight=2*ones(nx,1); % weight from iqcent+Diq3 and iqcent-Diq3
symweight(iqcent)=1; % centerpoint weight; allows limiting iq3 to half
the range

for iq1=iqmin:iqmax
for iq2=iqmin:iqmax

```

```

qsqr12=(iq1-iqcent)^2+(iq2-iqcent)^2;
for iq3=iqcent:iqmax

    % Channel sharing:
    % When qabs falls between two integer values, distribute intensity
    % between them

    qabs=sqrt(qsqr12+(iq3-iqcent)^2)+1; %|q| +1: prevent
    lscatt(iqabs=0)
    iqabs=round(qabs);
    ishar=qabs-iqabs; %
    share=abs(ishar); % how much to share
    isignshare=sign(ishar); % +1 if q > iq; 0 if exactly on

    addl=lq3D(iq1,iq2,iq3)*sincsqr(iq1)*sincsqr(iq2)*sincsqr(iq3)*symweig
    ht(iq3); %sincsqr: effect of elementary square

    lscatt(iqabs)=lscatt(iqabs)+addl*(1-share);
    lscatt(iqabs+isignshare)=lscatt(iqabs+isignshare)+addl*share; % 0 if
    exactly on

end
end
end

% -----
% calculate I(q)
% -----
for iq=2:nxfd2
    Iq(iq-1)=lscatt(iq)/(iq-1)^2; %iq=0 really q=0
end

```

```
clear Iq3
```

MatLab code of the algorithm depicted in Figure 1(b):

```
% Simplified version of SAXSrodcylfast.m  
% SAXS I(q) (orientationally averaged cross section) of parallel  
"infinite" rods/plates  
% from 2D density rho(x) of their cross sections
```

```
clear Ampl  
clear Iscatt  
clear Iq2D  
clear Iq  
clear sincsqr
```

```
nx=input('# of pts in each dimension (e.g. 2048)-->');  
nxf=input('# of q points in I(q) (e.g. 2048) -->');  
% can be smaller than nx, to speed up calculation  
% (gives only small-q results)  
nxd2=nx/2;  
nxfd2=nxf/2;  
iqcent=nxd2+1;  
  
%-----  
% From outside: scattering density rho (cross sections of cylinders)  
% make sure nx is the same in the external file  
%-----  
% -----
```

```

% Calculate scattering amplitude A(q) (complex)
% by Fourier transformation of rho(x)
% only a sheet (thin in q3) since rho is infinite along x3
% -----
Ampl=fft2(rho);
Ampl=fftshift(Ampl); %center of grid (nxd2+1) is at q=0

% ..... I(q)=|A(q)|^2 .....
Iq2D=abs(Ampl).^2;

%-----
% Prepare convolution with elementary square of area a^2
% by multiplication with sinc function in each dimension
%-----
iq1=(1:nx)-iqcent;
q1ad2=pi*iq1/nx+0.00000001; % q1*a/2 = iq1c*2pi/(nx*a) *a/2
sincsqr=(sin(q1ad2)./q1ad2).^2;

%-----
% Now calculate I(|q|)*q^2 by summing over all orientations
% this is I(q)*q^2, because the density of the cubic grid varies like q^2
%-----
Iscatt=zeros(nxf,1);

iqmin=iqcent-nxfd2;
iqmax=iqcent+nxfd2-1;

for iq1=iqmin:iqmax %
for iq2=iqmin:iqmax %
for iq3=iqcent:iqcent %thin sheet along q3

```

```

% Channel sharing:
% When qabs falls between two integer values, distribute intensity
between them

% To speed up the calculation, you could precalculate qabs and ishar
externally
qabs=sqrt((iq1-iqcent)^2+(iq2-iqcent)^2)+1; % +1: prevent
lscatt(iqabs=0)
iqabs=round(qabs);
ishar=qabs-iqabs;

share=abs(ishar); % how much to share
isignshare=sign(ishar); % +1 if q > iq; 0 if exactly on

addl=lq2D(iq1,iq2)*sincsqr(iq1)*sincsqr(iq2); %sincsqr: effect of
elementary square

lscatt(iqabs)=lscatt(iqabs)+addl*(1-share);
lscatt(iqabs+isignshare)=lscatt(iqabs+isignshare)+addl*share; % 0 if
exactly on

end
end
end

% -----
% calculate I(q)
% -----
for iq=2:nxfd2
Iq(iq-1)=lscatt(iq)/(iq-1)^2; %iq=0 really q=0

```

end