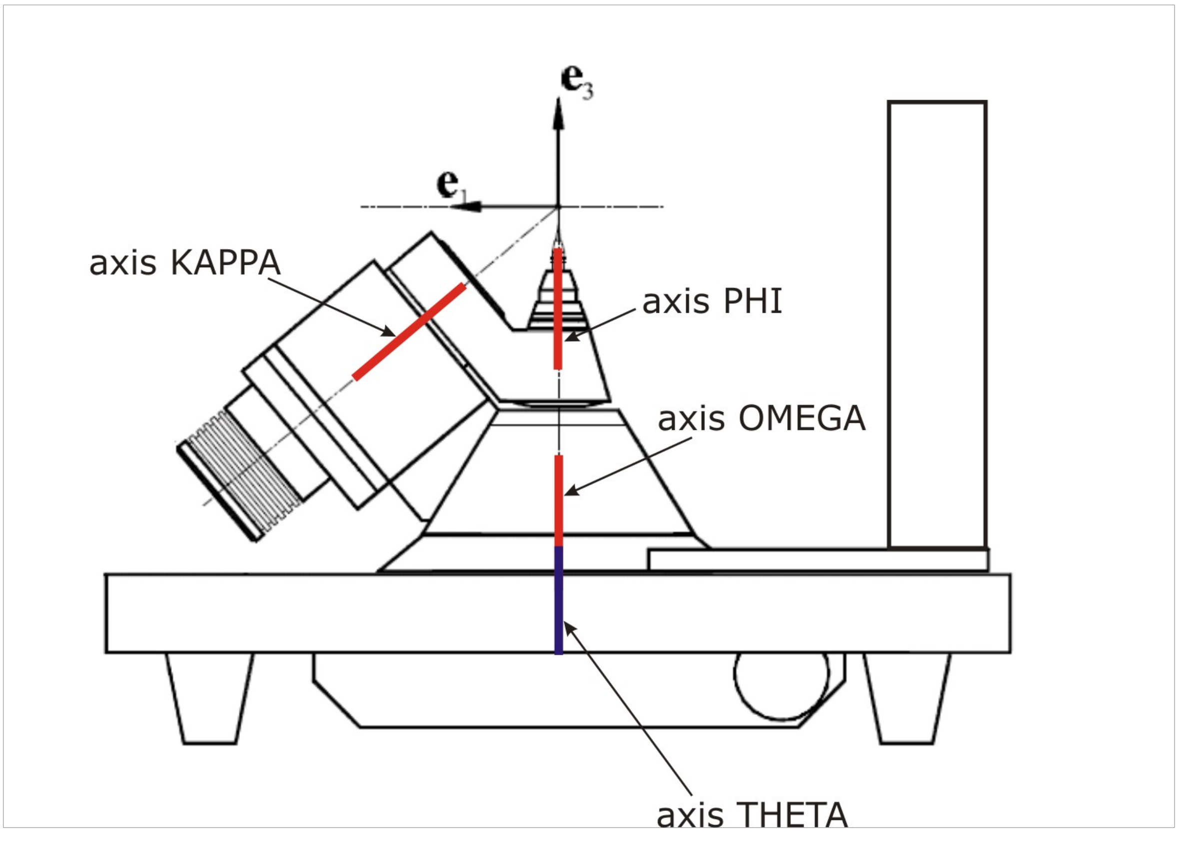
## Area detector data treatment (3D+2D)

Definitions for kappa-geometry as taken in *thesis_98_mathias.pdf*:

kappa axis has an angle $\alpha$ with $\mathbf{e}_3$

phi axis has an angle $\beta$ with $\mathbf{e}_3$

NB: usually $\beta = 0$ but in principle can be variable




$$R1(x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(x) & \sin(x) \\ 0 & -\sin(x) & \cos(x) \end{pmatrix}$$

$$R2(x) = \begin{pmatrix} \cos(x) & 0 & -\sin(x) \\ 0 & 1 & 0 \\ \cos(x) & 0 & \cos(x) \end{pmatrix}$$

$$R3(x) = \begin{pmatrix} \cos(x) & \sin(x) & 0 \\ -\sin(x) & \cos(x) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

**Primary rotation of reciprocal space** is described by

R = R3(OMEGA+O0)R2($\alpha$)R3(KAPPA)R2(-$\alpha$)R2($\beta$)R3(PHI)R2($\beta$)

where OMEGA, KAPPA and PHI are the nominal values of the angles and O0 is the OMEGA offset

For the arbitrary geometry of goniostat the rotation matrix will be generalized as R(angle1,...,angleN, const1,...,constM, offset1,...,offsetN)

*/generalization should be planned/*

**Rotation matrix for the detector** is

DET = R3(THETA+T0)R2(D2)R1(D1)

where THETA is the nominal theta value, T0 is the THETA offset, D1 and D2 are the tilts of detector

For the arbitrary geometry of goniostat the rotation matrix will be generalized as R(angle1,...,angle3, offset1,...,offset3)

*/generalization should be planned/*

**Secondary rotation of reciprocal space** to orient the crystallographic axis in a special way

U = R3(r3)R2(r2)R1(r1)

Definition in terms of Euler angles will be possible

*/generalization should be planned/*

**Beam tilt** matrix

B = R2(B2)

Can be generalized to

B = R3(B3)R2(B2)

*/generalization should be planned/*

**Primary projection of pixel coordinates (X,Y) to the reciprocal space**

Step 1: $p0 = B \begin{pmatrix} DIST \\ 0 \\ 0 \end{pmatrix}$

where DIST is the distance from the sample to the detector along the normal

Step 2: $\begin{pmatrix} x \\ y \end{pmatrix} = MD \cdot PIXELSIZE \cdot \begin{pmatrix} X - X0 \\ Y - Y0 \end{pmatrix}$

X0, Y0 – position of normal to the detector plane starting from the sample position,

PIXELSIZE is the size of pixel

MD – unit matrix by default, in a general case can be one of those:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$

*/generalization should be planned/*

Step 3: $p = DET \begin{pmatrix} -DIST \\ x \\ y \end{pmatrix}$

Step 4: $Q0 = \left( \frac{p}{|p|} + \frac{p0}{DIST} \right) \cdot \frac{1}{LAMBDA}$

Step 1 are performed once

steps 2, 3 and 4 are performed once for each THETA value

Q0(X,Y) is stored in memory

Calculating the corrections:

**Polarisation correction**

$$POL(X,Y) = p \left[ 1 - \left( \frac{(p0 \times n) \cdot p(X,Y)}{|p0 \times n||p(X,Y)|} \right)^2 \right] + (1-p) \left[ 1 - \left( \frac{n \cdot p(X,Y)}{|p(X,Y)|} \right)^2 \right]$$

n is the normal to the polarization plane (vector)

p is the degree of polarization

example (our starting configuration): for small offsets, p = 1 and n = $(0\ 0\ 1)^T$

$$POL(X,Y) = \frac{DIST^2 + y^2}{DIST^2 + x^2 + y^2}$$

**Flux density and parallax correction**

$$C3(X,Y) = \frac{DIST^3}{(DIST^2 + x^2 + y^2)^{3/2}}$$

*/generalization of C3(X,Y) should be planned/*

Corrections are calculated once for each THETA value

**Estimation of Qmax**

Except for the pathological cases, Qmax can be estimated taking the maximum of |Q0| over for corners of detector for all THETA settings

**Projection of given image to the reciprocal space**

$Q(X,Y) = R^{-1}Q0(X,Y) = R^{T}Q0(X,Y)$

here R depends on the goniometer angles (PHI, KAPPA, OMEGA or generalized)

NB: Q0 for given THETA must be used

**Orientation of given image in the reciprocal space**

$Qfin(X,Y) = U^{-1}Q(X,Y) = U^{T}Q(X,Y)$

**Constructing of 3D intensity distribution (non-optimized algorithm)**

create float32 array A set to zero values

create int32 array B set to zero values

for given image: intensity $\frac{I(X,Y)}{POL(X,Y)C3(X,Y)} \cdot mask(X,Y)$ is added to the corresponding element of A

*mask*(X,Y) is added to the same element of B

*mask*(X,Y) contains 1 for pixels to treat or 0 for pixels to skip; can be constant for all the images or depend on the image if the filtering is applied

For the image i the variable angle (i.e. PHI) is $(angle_i + angle_{i+1})/2$

NB: Large angular step or large magnification will create gaps between the projections of flat images to the reciprocal space; kind of interpolation is required.

The simplest solution: the same image is projected to the reciprocal space n times with finer angular step.

$angle(i,j) = angle_i + (j+1/2)(angle_{i+1} - angle_i)/n$, where j runs from 0 to n-1

default settings:

A and B are equivalent to $(2^N+1)\times(2^N+1)\times(2^N+1)$ cube

Qfin = 0 in the center of cube

cube size is $(2Qmax)\times(2Qmax)\times(2Qmax)$

User-defined settings

center of cube q0x, q0y, q0z

extent of sampling ±dqx, ±dqy, ±dqz

for the 3D array the indices will be

i = floor($2^{N-1}$[1+(Qxfin-q0x)/dqx])

j = floor($2^{N-1}$[1+(Qyfin-q0y)/dqy])

t = floor($2^{N-1}$[1+(Qzfin-q0z)/dqz])

if |Qxfin-q0x|≤dqx, |Qyfin-q0y|≤dqy, |Qzfin-q0z|≤dqz

Qxfin – x component of Qfin, Qyfin – y component of Qfin, Qzfin – z component of Qfin

normalize A by the counter B if the element of B is not zero

store A as binary array

store the restricted version ($2^N$)× ($2^N$)× ($2^N$) as CCP4 file (see description CCP4_format)

**Constructing of 2D intensity distribution (non-optimized algorithm)**

create float32 array A set to zero values

create int32 array B set to zero values

for given image: intensity $\frac{I(X,Y)}{POL(X,Y)C3(X,Y)} \cdot mask(X,Y)$ is added to the corresponding element of A

*mask*(X,Y) is added to the same element of B

default settings:

A and B are equivalent to square matrix (i.e. 2300×2300)

User-defined settings

orientation of cut: two vectors a and b’

check if they are not collinear

a = a/|a|

generate c = a×b’; c = c/|c|

generate b = c×a

$$G=\begin{pmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{pmatrix}$$

Qn = G•Qfin

offset vector Qoff

extent of sampling dQ1, dQ2

cut thickness ±dq

for the 2D array the indices will be

i = floor(size • [1+(Qxn-Qxoff)/(2dQ1)])

j = floor(size • [1+(Qyn-Qyoff)/(2dQ2)])

if |Qxn-Qxoff|≤dQ1, |Qyn-Qyoff|≤dQ2, |Qzn-Qzoff|≤dq

Qxn – x component of Qn, Qyn – y component of Qn, Qzn – z component of Qn

Qxoff – x component of Qoff, Qyoff – y component of Qoff, Qzoff – z component of Qoff

normalize A by the counter B if the element of B is not zero

store A in specified format

**EXPERIMENT DESCRIPTION PARAMETERS**

| | |
|---|---|
| ALPHA = $\alpha$ | factory setting |
| BETA = $\beta$ | factory setting |
| O0 | OMEGA offset |
| B2 | beam tilt |
| X0 | position of the normal from the sample on the detector |
| Y0 | position of the normal from the sample on the detector |
| T0 | THETA offset |
| R1 | orientation matrix angle |
| R2 | orientation matrix angle |
| R3 | orientation matrix angle |
| D1 | detector tilt |
| D2 | detector tilt |
| LAMBDA | wavelength |
| PIXELSIZE | size of pixel |
| DIST | distance sample-detector along the normal |
| THETA | image-specific THETA value |
| KAPPA | image-specific KAPPA value |
| OMEGA | image-specific OMEGA value |
| PHI | image-specific PHI value |

data series

**PARAMETERS FOR 3D RECONSTRUCTION**

| | |
|---|---|
| $2^N$ | final cube size |
| n | number of steps in the interpolation (starting from 1) |
| q0x, q0y, q0z | center of reconstruction cube |
| dqx, dqy, dqz | half-extent of reconstructed volume (±dqx, ±dqy, ±dqz) |
| output file name | |

**PARAMETERS FOR 2D RECONSTRUCTION**

| | |
|---|---|
| N | final square size |
| n | number of steps in the interpolation (starting from 1) |
| ax, ay, az | components of first vector |
| bx, by, bz | components of second vector |
| Qxoff, Qyoff, Qzoff | components of offset vector |
| dQ1, dQ2 | half-extent of the reconstructed cut |
| dq | half-thickness of integrated layer (±dq) |
| output file name | |

**EXTRACTION OF DESCRIPTORS FROM XCALIBUR PAR FILE**

see feo1_1.par

**ALPHA and BETA**

§ - ALPHA (DEG) 50.00000 BETA (DEG) 0.00000

**LAMBDA**

§ - WAVELENGTH USERSPECIFIED (ANG): A1 0.67018 A2 0.67018 B1 0.67018

Obsolete :

the following conversion was done at a time where data were etherogeneously converted from non-dectris setups, using a particular way of processing that then was needing the following transformation. In principle you could forget the following transformation and use X and Y out of the shelf, with a minor correction that is due to the fact that Crysalis converts internally the images to esperanto images that are square images, with padding and recentering. TDS2EL, in any case, is able to start from not so precise geometry and refine it.

**X0** = X – 1725 + vertical size

**Y0** = Y – 1725 + horisontal size

§ - DETECTOR ZERO (PIX, 1X1 BINNING): X 1733.19127 Y 1712.50841

**B2** = X2

§ - X-RAY BEAM ORIENTATION (DEG): X2 0.99075 X3 0.00000

**D1** = X1

**D2** = X2

§ - DETECTOR ROTATION (DEG): X1 -0.41144 X2 1.17097 X3 0.00000

**DIST** = DETECTOR DISTANCE (MM) • 1.72

§ - DETECTOR DISTANCE (MM): 174.42000

**O0** = OMEGA

**T0** = THETA

§ - SOFTWARE ZEROCORRECTION (DEG): OMEGA -0.19777 THETA 0.39804 KAPPA 0.00000 PHI 0.00000

**REMAINING PARAMETERS**

**KAPPA** = -134 imposed

**OMEGA** = 57 imposed

**THETA** = 0 imposed

**PHI** = (n-1) • 0.1 imposed, n is the number of image

**r1** = -88.788 imposed, in principle can be recovered from PAR file

**r2** = 2.257 imposed, in principle can be recovered from PAR file

**r3** = 69.629 imposed, in principle can be recovered from PAR file

**PIXELSIZE** = 0,172 mm