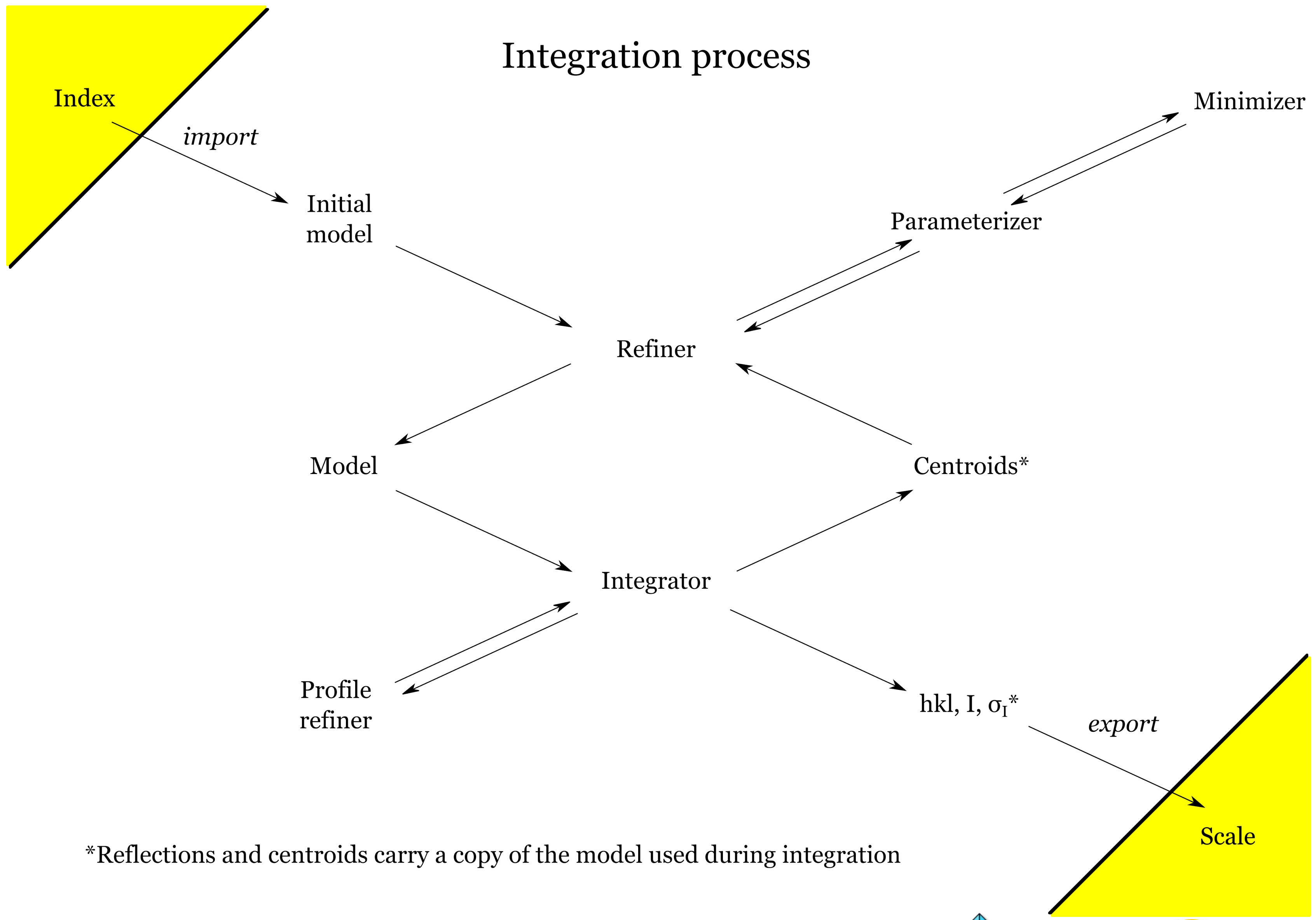
Diffraction geometry parameterisation and refinement

David Waterman
CCP4

Cambridge 22 May 2013

DIALS framework



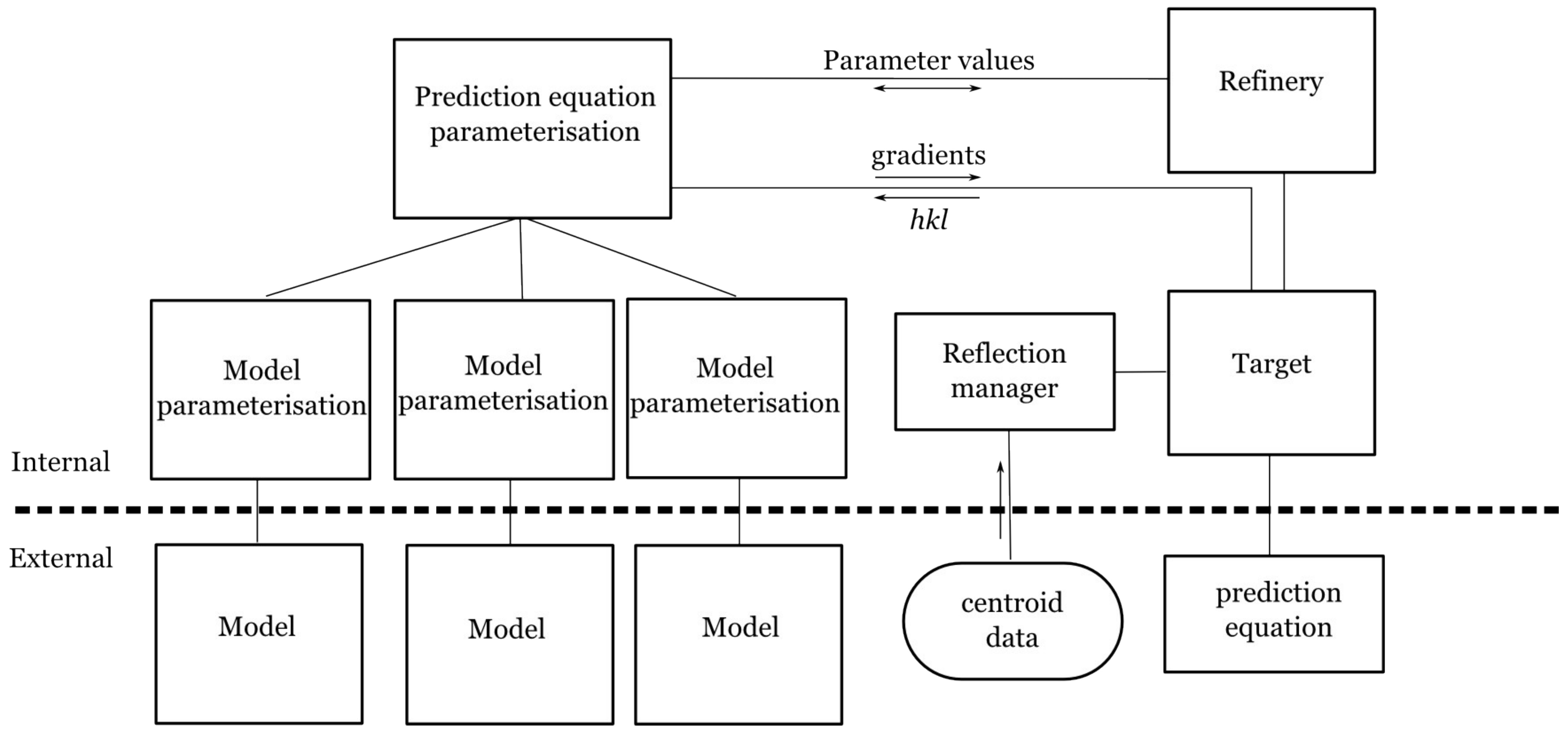
*Reflections and centroids carry a copy of the model used during integration

Motivation for global refinement

- Input diffraction spot indices, their centroids and estimated uncertainties ($h, k, l; X, Y, \phi; \sigma_X, \sigma_Y, \sigma_\phi$)
- Use all (useful) data available to refine a model to reduce rmsd of predicted centroids
- Global refinement helps to recover from poorly defined parameters in local ϕ window
- More physically meaningful: avoids mopping up of effects by correlated parameters and therefore obtains realistic parameter values
- Refine profile parameters separately
- Potential second round with improved centroid observations

Design

Overview



Progress to date

- April 2012
  - Start work on detector model and parameterisation
- After Algorithm Camp II (May 2012)
  - Broad overview of module design formed with Target-Minimiser-Basis organisation and collection of parameter values and derivatives from components of a global model
- June 2012
  - Abstract model parameterisation class and a detector parameterisation instance
- August 2012
  - Tested detector parameterisation derivatives by finite differences
  - First refinery class
  - First refinement of detector parameters by L-BFGS on simulated data

Progress to date

- Sept 2012
  - Parameterisation of source orientation
- Oct 2012
  - Begin prediction equation parameterisation class (for derivatives of the basis function)
  - crystal model
  - goniometer model
- Nov 2012: LMB meeting
  - First orientation refinement results presented. Couldn't refine detector and source simultaneously!
- Dec 2012
  - Fixed derivatives of X, Y wrt parameters that also affect phi
  - All derivatives tested by FD
  - Add curvatures (LSQ approximation) to L-BFGS minimiser
  - Only expose free parameters to the minimiser

Progress to date

- Jan 2013
  - Rescaled angle parameters to mrad. Much improved results with L-BFGS
- Feb 2013: LBNL workshop
  - 3* speed-up on refactoring, JMP's help with moving code to C++, and avoiding an unnecessary copy
  - Don't normalise target by number of preds each cycle (req changes to prediction to return a result even when out of Panel bounds)
  - Interface to NKS crystal unit cell parameterisation. Test all gradients.
  - Supply inverse of curvatures (i.e. diagonal of Hessian) to minimiser(!)
  - Move refinement code to new DIALS project on sourceforge

Progress to date

- Mar 2013
  - Add LSTBX engine for non-linear least squares refinement by Gauss-Newton iterations (much better)
  - Convert to using new DIALS models (dxtbx) and DIALS reflection prediction throughout
- April 2013
  - DIALS centroid refinement sprint. First use against real data
- May 2013
  - Time dependent parameterisation of the crystal (work in progress)

Target function

- Simple least squares target. No restraints terms added (yet)

$$L=\frac{1}{2}\sum_{h} w_{x,h}(X_c-X_o)^2+w_{y,h}(Y_c-Y_o)^2+w_{\phi,h}(\phi_c-\phi_o)^2$$

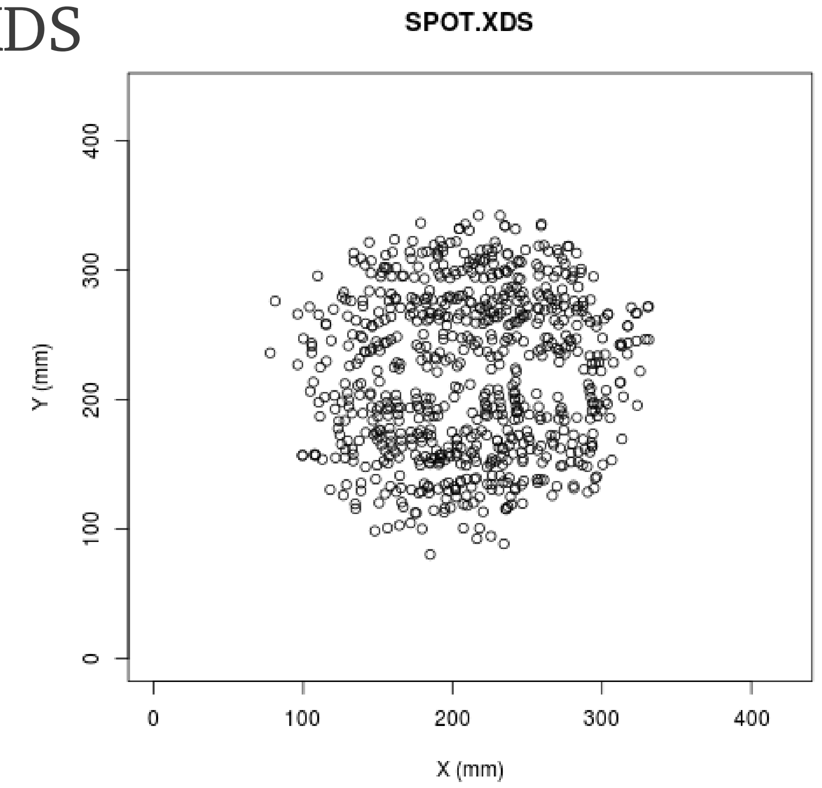
$$\frac{dL}{dp}=\sum_{h} w_{x,h}(X_c-X_o)\frac{dX_c}{dp}+w_{y,h}(Y_c-Y_o)\frac{dY_c}{dp}+w_{\phi,h}(\phi_c-\phi_o)\frac{d\phi_c}{dp}$$

- First order approximation to the curvatures

$$\frac{d^2L}{dp^2}\approx\sum_{h} w_{x,h}\left(\frac{dX_c}{dp}\right)^2+w_{y,h}\left(\frac{dY_c}{dp}\right)^2+w_{\phi,h}\left(\frac{d\phi_c}{dp}\right)^2$$

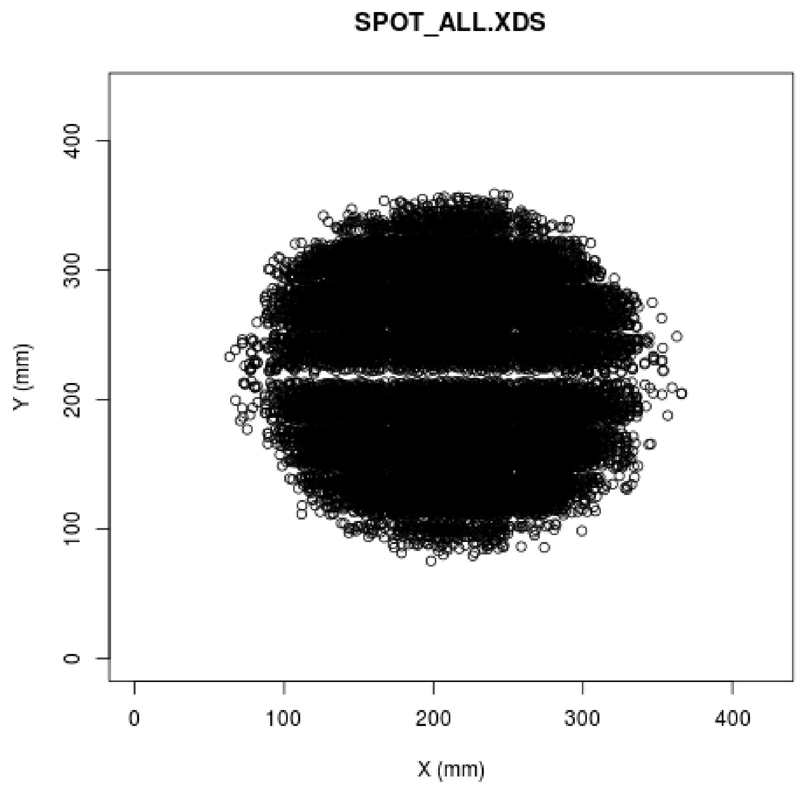
Centroid refinement results

- SPOT.XDS: 742 strong reflections distributed over three 0.8° wedges around 0°, 45° and 90°
- Refinement starting with GXPARM.XDS
  - gradient converged in 6 steps
  - RMSD_X slightly worse, RMSD_Y and RMSD_phi slightly better
- Refinement starting with XPARM_REGULARIZED.XDS
  - gradient converged in 8 steps
  - slightly worse rmsd than above



Centroid refinement results

- SPOT_ALL.XDS: 29023 reflections distributed over full 180° sweep
- Refinement starting with GXPARM.XDS
  - gradient converged in 3 steps
  - very slight improvement in rmsd
- Refinement starting with XPARM_REGULARIZED.XDS
  - gradient converged in 3 steps
  - obtains same end result



Centroid refinement results

```
Reading: "GXPARM.XDS"
Reading: "INTEGRATE.HKL"

Experimental Models
-------------------
Beam:
    wavelength: 0.9795
    direction : {0.00785262,-2.51934e-14,-0.999969}

Detector:
    Panel:
        type:          SENSOR_UNKNOWN
        fast axis:     {0.999956,0.00197257,0.00915725}
        slow axis:     {0.001983,-0.999997,-0.00113048}
        origin:        {-211.431,219.408,-192.801}
        normal:        {0.009155,0.00114859,-0.999957}
        pixel size:    {0.172,0.172}
        image size:    {2463,2527}
        trusted range: {0,0}

Goniometer:
    Rotation axis:  {1,-1.80647e-15,-8.38392e-15}
    Fixed rotation: {1,0,0,0,1,0,0,0,1}

Scan:
    image range:   {1,900}
    oscillation:   {0,0.2}
    exposure time: 0

Crystal:
    Unit cell: (42.275, 42.275, 39.669, 90.000, 90.000,
90.000)
    U matrix:  {{ 0.8336,  0.5360,  0.1335},
                {-0.1798,  0.0348,  0.9831},
                { 0.5223, -0.8435,  0.1254}}
    B matrix:  {{ 0.0237,  0.0000,  0.0000},
                { 0.0000,  0.0237,  0.0000},
                { 0.0000,  0.0000,  0.0252}}
    A = UB:    {{ 0.0197,  0.0127,  0.0034},
                {-0.0043,  0.0008,  0.0248},
                { 0.0124, -0.0200,  0.0032}}
```

```
Reading: "XPARM_REGULARIZED.XDS"
Reading: "INTEGRATE.HKL"

Experimental Models
-------------------
Beam:
    wavelength: 0.9795
    direction : {-0,1.22465e-16,-1}

Detector:
    Panel:
        type:          SENSOR_UNKNOWN
        fast axis:     {1,0,0}
        slow axis:     {0,-1,-1.22465e-16}
        origin:        {-212.754,219.609,-191.109}
        normal:        {0,1.22465e-16,-1}
        pixel size:    {0.172,0.172}
        image size:    {2463,2527}
        trusted range: {0,0}

Goniometer:
    Rotation axis:  {1,0,0}
    Fixed rotation: {1,0,0,0,1,0,0,0,1}

Scan:
    image range:   {1,900}
    oscillation:   {0,0.2}
    exposure time: 0

Crystal:
    Unit cell: (42.275, 42.275, 39.669, 90.000, 90.000,
90.000)
    U matrix:  {{ 0.8380,  0.5283,  0.1365},
                {-0.1808,  0.0328,  0.9830},
                { 0.5149, -0.8484,  0.1230}}
    B matrix:  {{ 0.0237,  0.0000,  0.0000},
                { 0.0000,  0.0237,  0.0000},
                { 0.0000,  0.0000,  0.0252}}
    A = UB:    {{ 0.0198,  0.0125,  0.0034},
                {-0.0043,  0.0008,  0.0248},
                { 0.0122, -0.0201,  0.0031}}
```

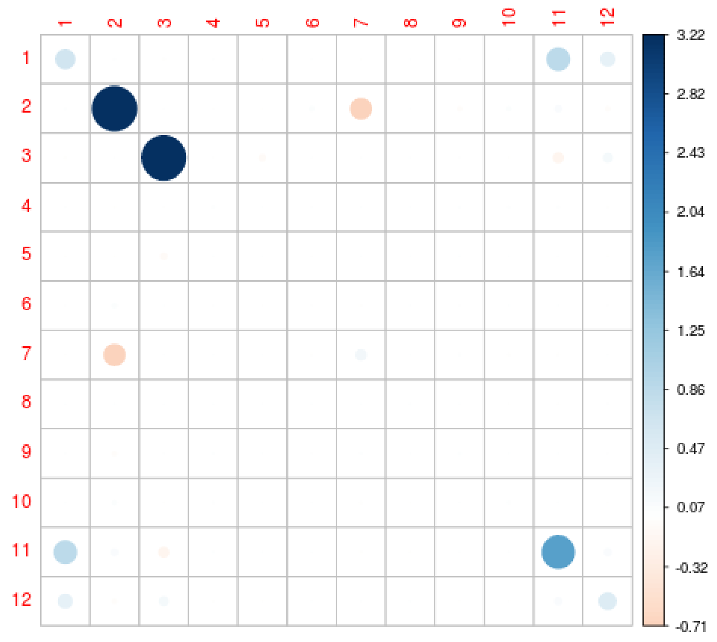
Centroid refinement results

- Refinement benefits from inclusion of more data throughout the sweep
- Need time-dependent crystal model to reduce RMSDs further
- When wrong parameter fixed, crash with “Cholesky error” → non-positive-definite N
- Will be useful to study properties of the normal matrix

Parameter correlation

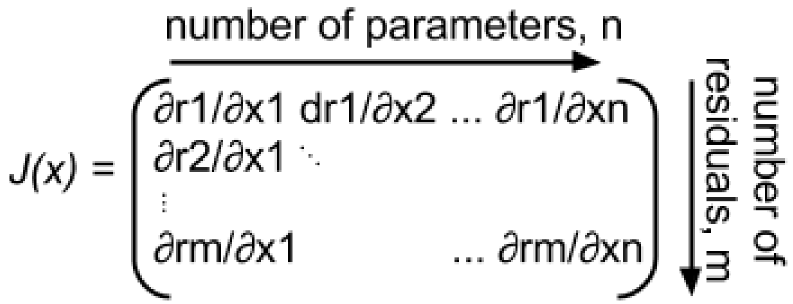
- Analysis of normal matrix. Not immediately insightful. 3 orders of magnitude range along diagonal
- The eigenvalues of N also range over a factor of 3000
- No reason to think this is “too much” though

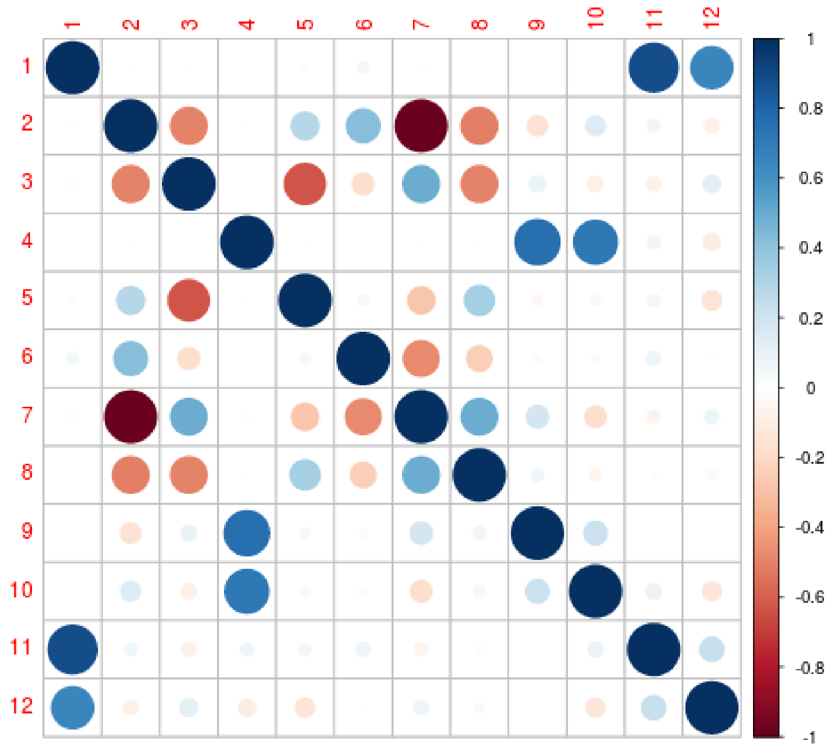
```
Parameter order:name mapping
Parameter 001 : DetectorDist
Parameter 002 : DetectorShift1
Parameter 003 : DetectorShift2
Parameter 004 : DetectorTau1
Parameter 005 : DetectorTau2
Parameter 006 : DetectorTau3
Parameter 007 : SourceMu2
Parameter 008 : CrystalPhi1
Parameter 009 : CrystalPhi2
Parameter 010 : CrystalPhi3
Parameter 011 : Crystal_g_param_0
Parameter 012 : Crystal_g_param_1
```



Parameter correlation

- Analysis of Jacobian J(x), where $N = J(x)^T J(x)$
- Can J be near rank-deficient?
- Easier to calculate correlation between columns of J
- Is J ill conditioned? I don't know





```
Parameter order:name mapping
Parameter 001 : DetectorDist
Parameter 002 : DetectorShift1
Parameter 003 : DetectorShift2
Parameter 004 : DetectorTau1
Parameter 005 : DetectorTau2
Parameter 006 : DetectorTau3
Parameter 007 : SourceMu2
Parameter 008 : CrystalPhi1
Parameter 009 : CrystalPhi2
Parameter 010 : CrystalPhi3
Parameter 011 : Crystal_g_param_0
Parameter 012 : Crystal_g_param_1
```

Extensions to LSTBX

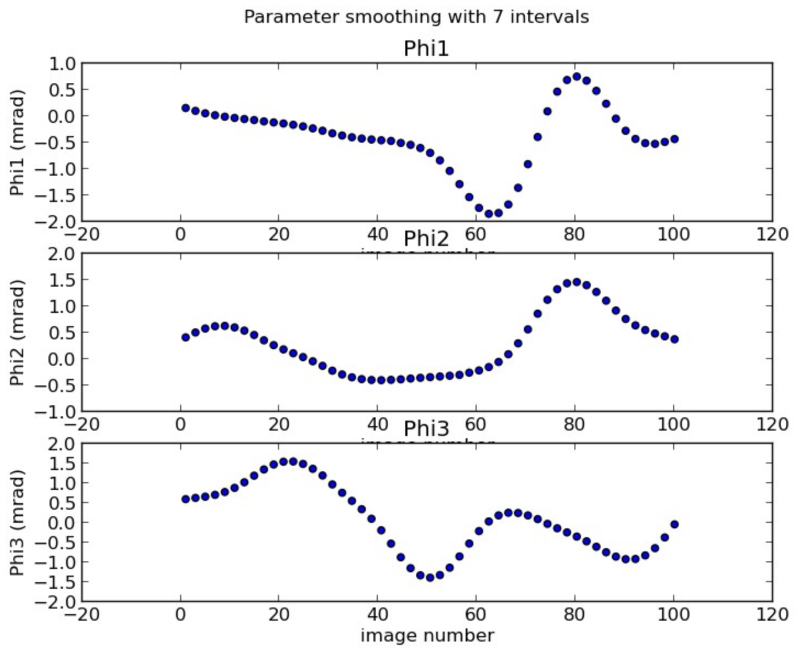
- LSTBX solves the normal equations using the Cholesky decomposition
- This is fast, but poorly behaved when J is ill-conditioned (accuracy suffers, or algorithm can even fail due to roundoff errors)
- QR is more robust, SVD even more so, at the expense of more CPU cycles
- SVD has the advantage of providing useful sensitivity information and option to filter out the smallest singular values to obtain an approximate solution less sensitive to perturbations

Extensions to LSTBX

- This appears closely related to the Reeke/Bricogne "eigenvalue filtering" scheme
- What is the "best" way to solve the normal equations, perhaps admitting the possibility of filtering for correlated parameters?
- Can we get error estimates on the parameter values even in the case of filtering?
- Plan to modify LSTBX to implement a procedure for solving the normal equations that is appropriate for our circumstances
- Also try Levenberg-Marquardt iterations rather than Gauss-Newton (already available in LSTBX) for better behaviour when J(x) is nearly rank-deficient

Time-dependent parameterisation

- Implemented Gaussian smoother from Aimless
- Derivatives $\partial\mathbf{U}(t)/\partial \mathrm{p}$ and $\partial\mathbf{B}(t)/\partial \mathrm{p}$ tested by FD for crystal orientation and unit cell parameters
- There are three adjustable variables for the smoother
  - number of samples
  - sigma
  - number of points to average
- How do we know what values are appropriate?



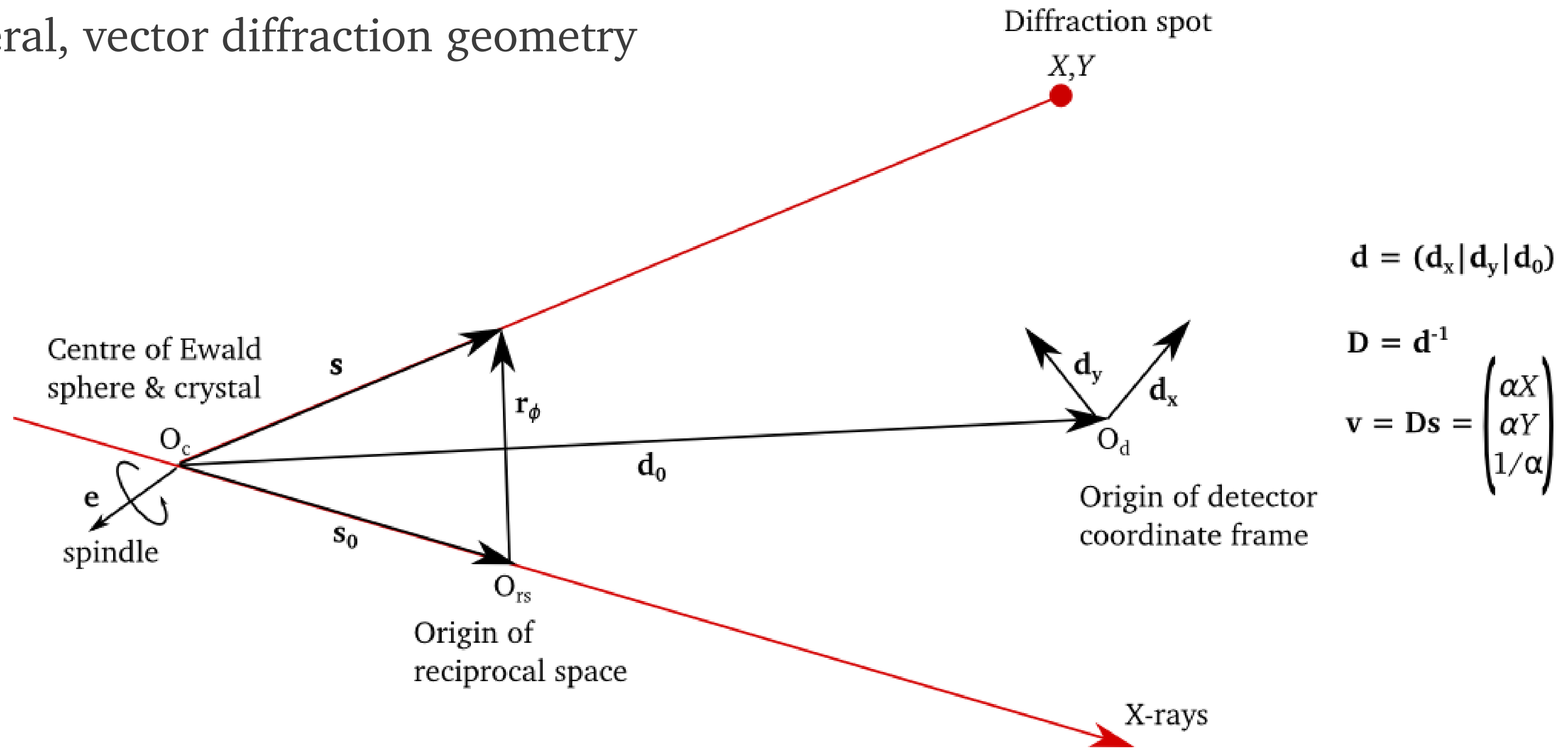
Time-dependent parameterisation

Proposed scheme for refinement:

- A fully time invariant macrocycle to convergence to improve the detector and source models and define $\mathbf{U}_0$ and $\mathbf{B}_0$
- A macrocycle using time-dependent crystal parameterisations and static detector and source parameterisations
  - Parameters of the time dependent (Gaussian smoothed) models are restrained (tied) to the values that define $\mathbf{U}_0$ and $\mathbf{B}_0$
- Integration forms models for profiles, potentially improving the centroid positions
- Repeat

Vectorial reflection prediction

- General, vector diffraction geometry



Diffraction spot
X,Y
$\mathbf{d} = (\mathbf{d}_x|\mathbf{d}_y|\mathbf{d}_0)$
$\mathbf{D} = \mathbf{d}^{-1}$
$\mathbf{v} = \mathbf{D}\mathbf{s} = \begin{pmatrix}\alpha X\\ \alpha Y\\ 1/\alpha\end{pmatrix}$
Centre of Ewald sphere & crystal
$\mathbf{s}$
$\mathbf{r}_\phi$
$\mathbf{d}_y$
$\mathbf{d}_x$
O_c
O_d
$\mathbf{d}_0$
$\mathbf{e}$
spindle
$\mathbf{s}_0$
Origin of detector coordinate frame
O_{rs}
Origin of reciprocal space
X-rays

- The detector abstract frame is a hardware-independent adapter
- Positional corrections can be accounted for in the mm-to-px mapping function

dials Vectorial reflection prediction

For refinement we want at least the first derivatives of predicted centroids

$$\frac{\partial\phi}{\partial p} = -\frac{\frac{\partial \mathbf{r}_\phi}{\partial p}\cdot\mathbf{s} + \mathbf{r}_\phi\cdot\frac{\partial \mathbf{s_0}}{\partial p}}{(\mathbf{e}\times\mathbf{r}_\phi)\cdot\mathbf{s_0}}$$

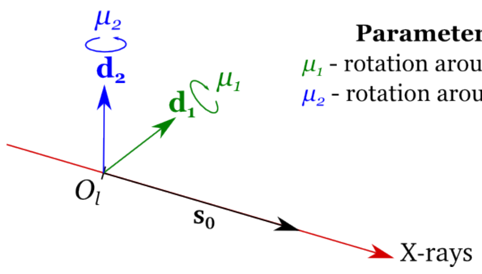
$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}p} = -\mathbf{D}\frac{\partial\mathbf{d}}{\partial p}\mathbf{v} + \mathbf{D}\left[\frac{\partial \mathbf{r}_\phi}{\partial p} + (\mathbf{e}\times\mathbf{r}_\phi)\frac{\partial\phi}{\partial p} + \frac{\partial \mathbf{s_0}}{\partial p}\right]$$

Neatly, these are factored into independent models

$\frac{\partial\mathbf{d}}{\partial p}$	$\frac{\partial \mathbf{r}_\phi}{\partial p}$	$\frac{\partial \mathbf{s_0}}{\partial p}$
detector	crystal	beam direction

Parameterisation

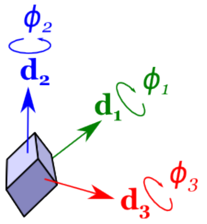
Each model parameterisation provides ∂[state]/∂p



Parameters
μ_1 - rotation around $\mathbf{d}_1$
μ_2 - rotation around $\mathbf{d}_2$

$\frac{\partial \mathbf{s}_0}{\partial p}$

beam direction



Parameters
ϕ_1 - rotation around $\mathbf{d}_1$
ϕ_2 - rotation around $\mathbf{d}_2$
ϕ_3 - rotation around $\mathbf{d}_3$
$a, b, c, \alpha, \beta, \gamma$ - unit cell

$\frac{\partial \mathbf{U}}{\partial p}$

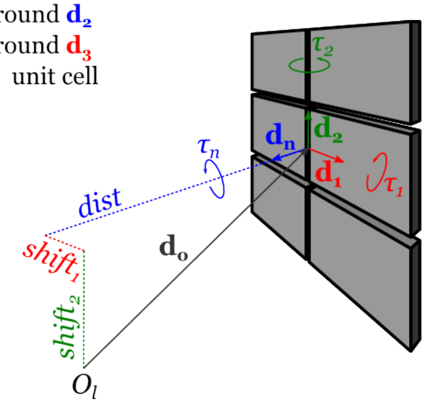
crystal orientation

$\frac{\partial \mathbf{B}}{\partial p}$

crystal unit cell

$\frac{\partial \mathbf{d}}{\partial p}$

detector position and orientation

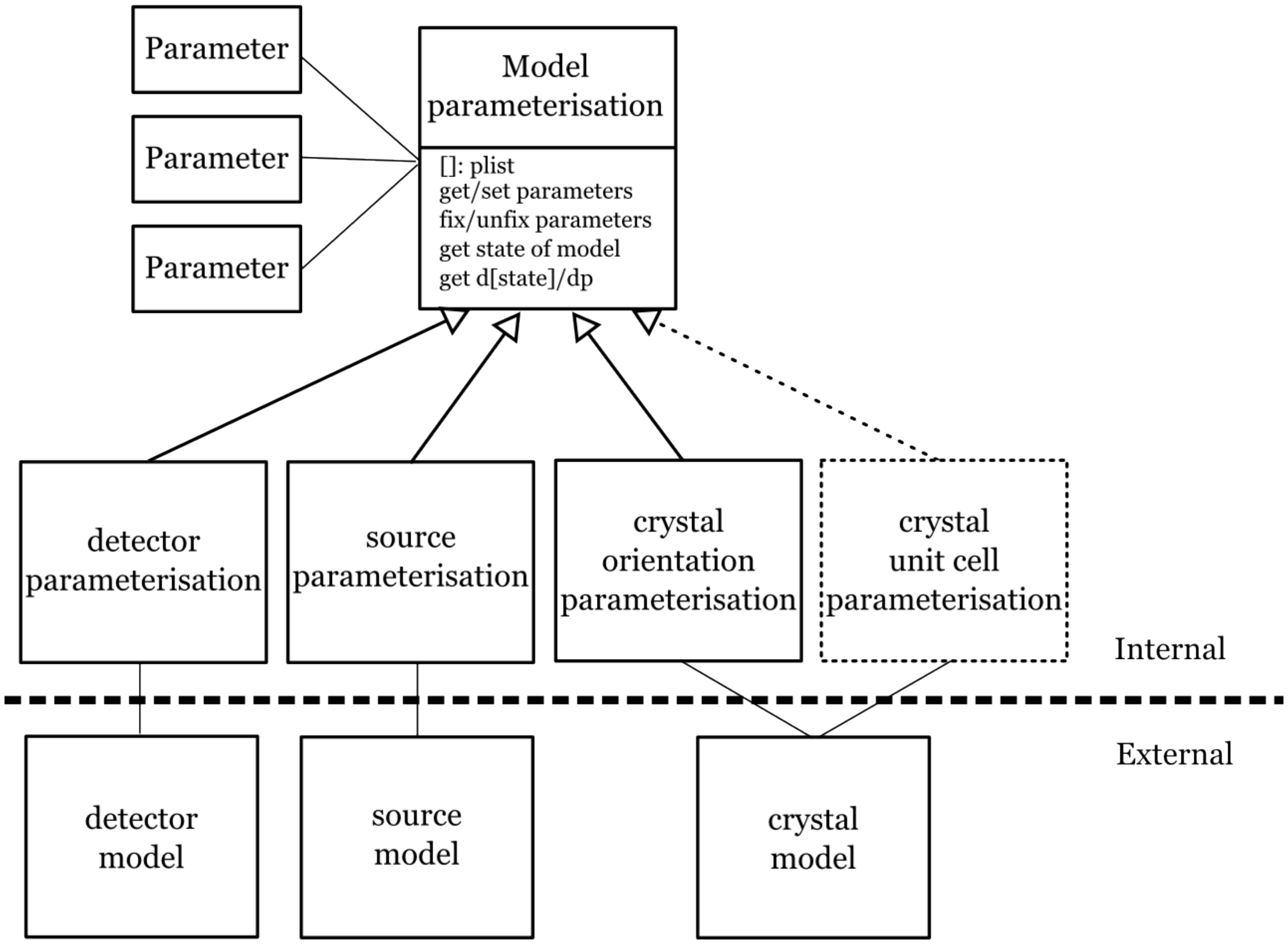


Parameters
dist - detector distance along $\mathbf{d}_n$
shift$_1$ - translation in direction $\mathbf{d}_1$
shift$_2$ - translation in direction $\mathbf{d}_2$
τ_n - rotation around $\mathbf{d}_n$ ≅ "roll"
τ_1 - rotation around $\mathbf{d}_1$ ≅"tilt"
τ_2 - rotation around $\mathbf{d}_2$ ≅"twist"

- Separate 'parameterisation of prediction equation' object takes ∂[state]/∂p for each model and converts to derivatives of X, Y, ϕ for each reflection
- Individual model parameterisations are encapsulated

Design

Parameterisation of experimental models



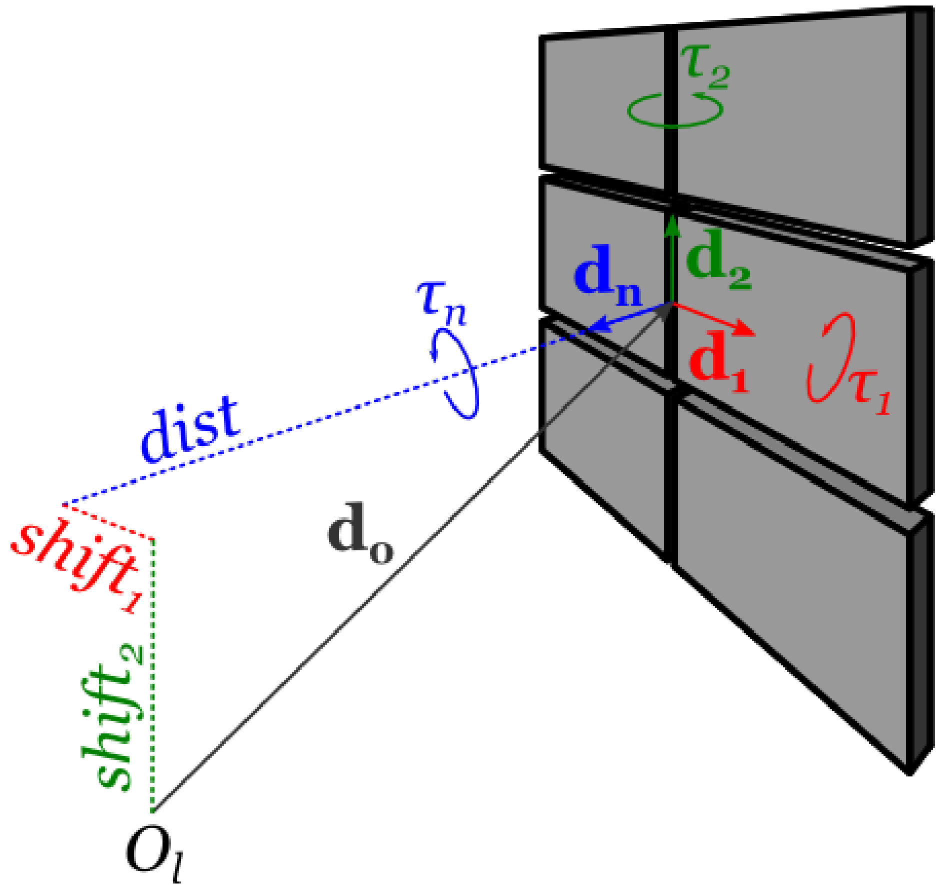
Parameterisation

The abstract interface specifies that:

- Model parameterisations are initialised with an initial state of the model
- New states are composed by the action of functions of the parameters on the initial state
- A state and its derivatives are either a vector or a matrix
- The parameters are either distances or angles with associated unit directions

Parameterisation

- A concrete example: detector parameterisation



Parameters

dist - detector distance along $\mathbf{d_n}$

$shift_1$ - translation in direction $\mathbf{d_1}$

$shift_2$ - translation in direction $\mathbf{d_2}$

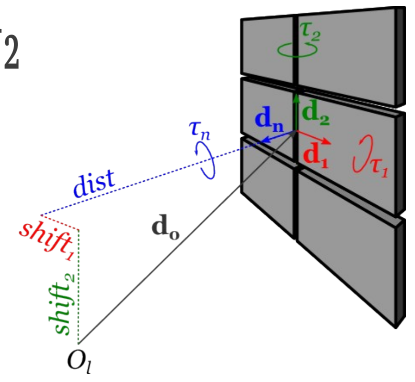
τ_n - rotation around $\mathbf{d_n}$ $\cong$ "roll"

τ_1 - rotation around $\mathbf{d_1}$ $\cong$"tilt"

τ_2 - rotation around $\mathbf{d_2}$ $\cong$"twist" dials

Parameterisation

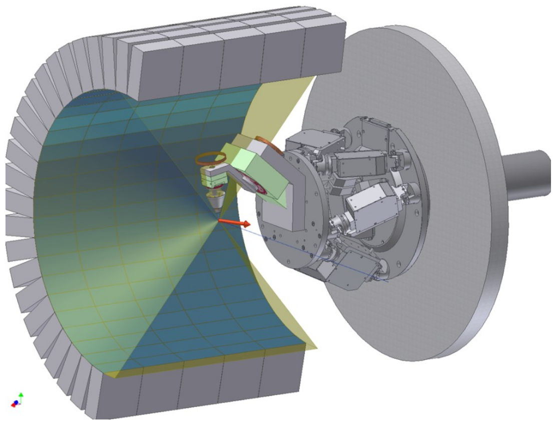
- Initial sensor matrix provides $\mathbf{d}_0$, $\mathbf{d}_1$, $\mathbf{d}_2$, $\mathbf{d}_n$
- Translation parameters are immediately *dist* along $\mathbf{d}_n$ and $shift_1$, $shift_2$ along $\mathbf{d}_1$, $\mathbf{d}_2$
- Initial rotation angles all 0.0, around axes $\mathbf{d}_1$, $\mathbf{d}_2$, $\mathbf{d}_n$



CCP4 diamond

Parameterisation

- Accommodates refinement of multi-tile detectors as one rigid unit
- Each sensor panel k has its own matrix $\mathbf{d}^k = (\mathbf{d_x}^k|\mathbf{d_y}^k|\mathbf{d_0}^k)$
- These vectors are linear combinations of $\mathbf{d_0}$ and the local coordinate system $\mathbf{d_1}$, $\mathbf{d_2}$, $\mathbf{d_n}$ that moves with the detector:



$$\mathbf{d}_x^{\,k} = \alpha_1^{\,k}\mathbf{d_1} + \alpha_2^{\,k}\mathbf{d_2} + \alpha_3^{\,k}\mathbf{d_n}$$

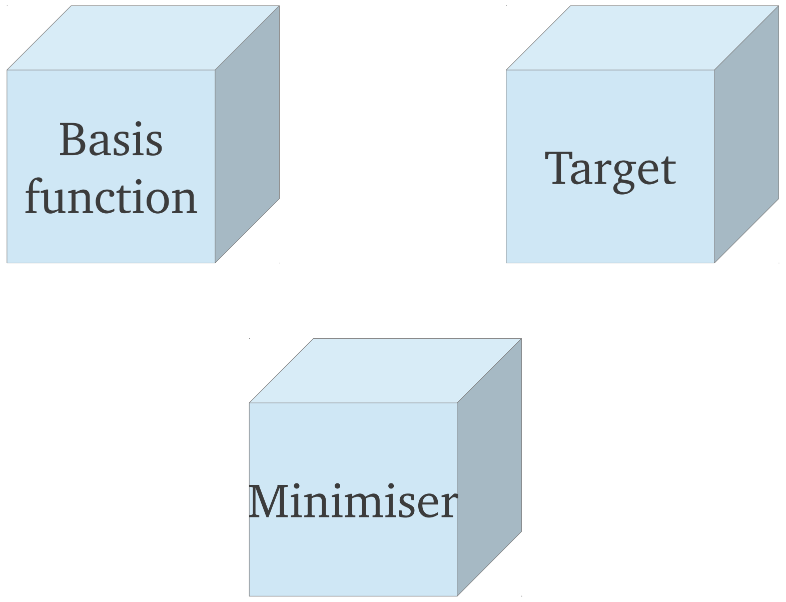
$$\mathbf{d}_y^{\,k} = \beta_1^{\,k}\mathbf{d_1} + \beta_2^{\,k}\mathbf{d_2} + \beta_3^{\,k}\mathbf{d_n}$$

$$\mathbf{d}_0^{\,k} = \mathbf{d_0} + \gamma_1^{\,k}\mathbf{d_1} + \gamma_2^{\,k}\mathbf{d_2} + \gamma_3^{\,k}\mathbf{d_n}$$

- Thus the derivatives $\partial\mathbf{d}^k/\partial p$ for each sensor are easily calculated by linear combinations of $\partial\mathbf{d_o}/\partial p$, $\partial\mathbf{d_1}/\partial p$, $\partial\mathbf{d_2}/\partial p$ and $\partial\mathbf{d_n}/\partial p$

Design

- Further encapsulation within refinement module



- Make these independent (where possible)