Coordinate Systems

Coordinate systems, operators, and transformations.

Kevin Cowtan
cowtan@ysbl.york.ac.uk

Coordinate Systems

- In addition, we need to handle:
  - Transformations between these coordinate systems.
  - Transformations within a coordinate system (i.e. rotation and translation operators).
  - Rotation representations.
  - Derivatives of functions with respect to different coordinate systems.
- This lecture will give a basic overview of the issues. Implementations of all these data types and transformations are a part of both the CCTBX and Clipper crystallographic libraries.

Coordinate Systems

- Conventions for this lecture:
  - Scalars are italic, lower case, e.g. $s$, $r$
  - Vectors are italic, lowercase and underlined, e.g. $\mathbf{v}$, $\mathbf{h}$
  - Matrices are italic, uppercase and bold, e.g. $\mathbf{O}$, $\mathbf{F}$, $\mathbf{M}$
- In addition to matrix notation, most equations are also given as explicit sums of terms.
  - The elements of a vector or matrix are given by the same symbol in italic lower case with an appropriate number of subscripts. e.g. $x_i$, $O_{ij}$.
- All vectors and matrices are of rank 3.

Coordinate Systems: Real space

Orthogonal Ångstrom Coordinates:

- 3 orthogonal distances in Ångstroms along directions $x$, $y$, $z$ (Formally: basis vectors)

$$ x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} $$

Note: there is no reference to the unit cell at this point.
Coordinate Systems: Real space

Fractional coordinates:
- Position in the unit cell described as a fractional position along each cell edge:

\[ \mathbf{u} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \]

Note: since the cell repeats, u,v,w repeat on the range 0...1. We often standardize on the range 0...1 (or -1/2...1/2), but this may split a molecule.

Relating orthogonal and fractional coordinates:
- We can orient and position one coordinate system however we want with respect to the other, but...
- It is convenient to adopt some convenient convention.
- The most common convention in the PDB (also the CCP4 default) is:
  - Align the \( a \) axis along \( x \)
  - Align the \( b \) axis in the \( xy \) plane
  - Or equivalently align \( z \) axis perpendicular to \( a \) and \( b \)

An orthogonal coordinate may be determined from a fractional coordinate by:
\[ x_i = \Sigma_j O_{ij} u_j \]

Where \( O \) is the orthogonalization matrix. For the common convention,
\[ O = \begin{pmatrix} a & b \cos(y) & c \cos(\beta) \\ 0 & b \sin(y) & -c \sin(\beta) \cos(\alpha^*) \\ 0 & 0 & c \sin(\beta) \sin(\alpha^*) \end{pmatrix} \]

A fractional coordinate may be determined from an orthogonal coordinate by:
\[ u_j = \Sigma_i F_{ij} x_i \]

Where \( F \) is the fractionalization matrix.

Measuring distances:
- We do this all the time, e.g. inter-atomic distances.
- In orthogonal coordinates, the squared distance between two points is given by \( r^2 = \Delta x^2 + \Delta y^2 + \Delta z^2 \)

\[ r^2 = \Sigma_i \Delta x_i^2 \]
Coordinate Systems: Real space

Measuring distances:

- For the distance between two fractional coordinates, convert to orthogonal first:
  \[ r^2 = \Delta u^T O^T O \Delta u \]
  or:
  \[ r^2 = \Sigma_i \Sigma_j \Sigma_k O_{ij} O_{ik} \Delta u_j \Delta u_k \]

- Simplify by pre-calculation of the central product:
  \[ M = O^T O \]
  \[ M_{jk} = \Sigma_i O_{ij} O_{ik} \]
- \( M \) is a symmetric matrix, called the "real-space metric tensor".

Coordinate Systems: Real space

Simplified form using the metric tensor:

\[ r^2 = M_{11} \Delta u_1^2 + M_{22} \Delta u_2^2 + M_{33} \Delta u_3^2 \]
\[ + 2 M_{12} \Delta u_1 \Delta u_2 + 2 M_{13} \Delta u_1 \Delta u_3 + 2 M_{23} \Delta u_2 \Delta u_3 \]

(since the matrix is symmetric, we just use the upper triangle and double the off-diagonal terms).

(This is often a performance critical task).

Coordinate Systems: Real space

Other coordinate types:

- Grid coordinates: Electron density maps are usually calculated on a grid which samples the unit cell.

Grid coordinates: Additional complications:

- Hexagonal close packed grids give a more efficient sampling of real space. How are they best indexed?
Coordinate Systems: Real space

Other coordinate types:

- Map coordinates: (Cowtan)
  - Non-integer grid coordinates.
    - For crystallographic maps, fractional coordinates do the job just fine.
    - For non-crystallographic maps, fractional coordinates are undefined.
  - Used for interpolation in non-crystallographic maps.

Coordinate Systems: Real space

Implementation issues:

- It is very easy when programming to make mistakes over coordinate types.
- When using strongly typed languages (e.g. C++), implement each coordinate type as a different class to prevent such errors.
  - (Use inheritance for common behaviors)
- Coordinate types or cell and sampling classes may then implement all the required conversions.
  See CCTBX or Clipper.

Coordinate Systems: Reciprocal space

In reciprocal space we mainly deal with reflections, indexed by integer $h,k,l$:

$h = \begin{pmatrix} h \\ k \\ l \end{pmatrix} = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix}$

$h,k,l$ are coordinates on a non-orthogonal grid, like grid coordinates.
$h,k,l$ do not repeat. They are centered (+/-) about the origin $h=k=l=0$.
May be referred to as Miller indices (by correspondence with the indexing of crystal cell faces).

Coordinate Systems: Reciprocal space

We frequently need to determine the (reciprocal) distance from a reflection to the origin (resolution).
We may also need to determine a reciprocal orthogonal coordinate e.g. to make this picture.

Coordinate Systems: Reciprocal space

Relating reciprocal orthogonal and fractional coordinates:

- An reciprocal orthogonal coordinate may be determined from an HKL by:

$\mathbf{s} = \mathbf{O}^* \cdot \mathbf{h}$

- As in real space, $\mathbf{O}^*$ is determined by the orthogonalization convention. A convenient choice is to use the transpose of the real space fractionalizing matrix: $\mathbf{O}^* = F^T$
  - i.e. $\mathbf{z}$ parallel to $\mathbf{z}^*$, $\mathbf{z}$ in the $\mathbf{b}^* \cdot \mathbf{c}^*$ plane
Coordinate Systems: Reciprocal space

Relating reciprocal orthogonal and fractional coordinates:

- For display purposes, the “Cambridge Convention” is more common:
  - \( \mathbf{x} \) parallel to \( \mathbf{a}^* \)
  - \( \mathbf{y} \) in the \( \mathbf{a}^* \mathbf{b}^* \) plane

- In this case \( \mathbf{O}^* \) is calculated using the equivalent formula to \( \mathbf{O} \).

- Don’t mix conventions between real and reciprocal space – it will only end in tears.

Coordinate Systems: Reciprocal space

Measuring distances in reciprocal space:

- As before, to calculate squared distances (in inverse squared Ångstroms), we first need reciprocal orthogonal coordinates, or a reciprocal metric tensor:
  \[
  s^2 = h^T \mathbf{O}^* \mathbf{O}^* h
  \]
  or:
  \[
  s^2 = \sum_i \sum_j \sum_k \mathbf{O}^*_{ij} \mathbf{O}^*_{ik} h_i h_j h_k
  \]

- Simplify by pre-calculating the central product:
  \[
  \mathbf{M}^* = \mathbf{O}^* \mathbf{O}^*
  \]
  \[
  \mathbf{M}^*_{jk} = \sum_i \mathbf{O}^*_{ij} \mathbf{O}^*_{ik}
  \]

  \[
  s^2 = \mathbf{h}^T \mathbf{M}^* \mathbf{h}
  \]

Resolution in Ångstroms is \( 1/s \), i.e. \( \sqrt{1/s^2} \).

Some developers use the symbol \( s \) instead of \( s^2 \).

Clipper and CCP4 refer to \( s^2 \) as inverse resolution squared: “invresolsq”.

Interlude

The Structure Factor Equation:

- From Scattering Theory:
  \[
  F(s) = \sum_j f_j(s) \exp(2\pi i s^T \mathbf{x})
  \]

- Structure Factor Equation:
  \[
  F(h) = \sum_j f_j(s) \exp(2\pi i h^T \mathbf{u})
  \]

Because:

- \( s^T \mathbf{x} = (F^T \mathbf{h})^T (\mathbf{O} \mathbf{u}) \)
- \( = h^T \mathbf{F O} \mathbf{u} \)
- \( = h^T \mathbf{u} \)

Coordinate Systems: Operators

Operators transform coordinates in such a way that a rigid body will moved to a new position within the coordinate system. We consider three types:

- Translation operators:
  - Move an object without rotating it.

- Rotation operators:
  - Rotate an object about the origin of the coordinate system.

- Rotation-translation (RT) operators:
  - Rotate and translate an object, or equivalently, rotate an object about a point other than the origin.
Coordinate Systems: Operators

Rotation operators:

- A rotation is described by a matrix $R$ which is orthonormal (i.e. $R^{-1} = R^T$)

\[
R = \begin{bmatrix}
\cos(x_1 x_2) & \cos(x_1 y_2) & \cos(x_1 z_2) \\
\cos(y_1 x_2) & \cos(y_1 y_2) & \cos(y_1 z_2) \\
\cos(z_1 x_2) & \cos(z_1 y_2) & \cos(z_1 z_2)
\end{bmatrix}
\]

\[
x_2 = R x_1
\]

Since $x_1 = O u_1$ and $x_2 = O u_2$, 

\[
O u_2 = R u_1
\]

therefore:

\[
R_u = O^{-1} R_u O
\]

and:

\[
R_x = O R_u O^{-1}
\]

- Note: $R_u$ is not a rotation matrix. ($R_u^{-1} \neq R_u^T$)

Coordinate Systems: Operators

Rotation-translation (RT) operators:

- A rotation-translation operator is described by a rotation matrix $R$ followed by a translation $T$:

\[
x_2 = R x_1 + T_x
\]

We can represent this as a single vector operator:

\[
x_2 = R_i (x_i)
\]

Its inverse is given by the rotation $R_i^{-1}$ and the translation $-R_i^{-1} T_x$:

\[
x_1 = R_i^{-1} (x_2 - T_x) = R_i^{-1} x_2 - R_i^{-1} T_x
\]

- To convert to fractional, convert $R$ and $T$ as before.

Coordinate Systems: Rotations

Rotations have many representations:

- Matrix:
  - use directly to manipulate vectors
  - uniquely defined, 9 numbers ($m_{ij}$).

- Quaternions:
  - uniquely defined, 4 numbers ($x,y,z,w$).

- Euler angles:
  - multiple conventions, 3 numbers ($\alpha,\beta,\gamma$).

- Polar angles:
  - multiple conventions, 3 numbers ($\phi,\psi,\kappa$).

Quaternions:

- 4 numbers: $(x,y,z,w)$
  - $x,y,z$ are the direction cosines of the rotation axis, scaled by $\sin(\kappa/2)$
  - $w$ gives the angle of rotation, as $\cos(\kappa/2)$.

- No ambiguity in definition.

- Easy to convert to Matrix, Euler, Polar
  - good as an interchange format. See Clipper, rotation.cpp
Coordinate Systems: Rotations

Euler angles:

- 3 numbers: \((\alpha, \beta, \gamma)\).
  - \(\alpha\) is rotation about \(z\).
  - \(\beta\) is rotation about new \(y\).
  - \(\gamma\) is rotation about new \(z\).
- 24 conventions (which axis to rotate about, stationary or moving axes), but crystallographers all use \(ZYZ\), so well standardised.
- Convenient for rotation function search limits.
- Convenient for program input.

Polar angles:

- 3 numbers: \((\phi, \psi, \kappa)\) or \((\omega, \phi, \kappa)\)
  - \((\phi, \psi)\) define the direction of the axis, \(\kappa\) is the angle of rotation about it.
  - Easy to understand.
  - Inconsistent conventions.
  - Use for program output only.

Coordinate Systems: Derivatives

Many calculations require that we calculate derivatives of some function with respect to some coordinate. E.g.

- Refinement:
  - Refinement of individual atomic coordinates and B-factors: \((x_i, B_i)\)
- Molecular replacement:
  - Rigid body refinement of search model against density: \((R_x, T_x)\)

- e.g. for density gradients, \(f = \rho\)

\[
\frac{\partial f}{\partial u} = \begin{pmatrix}
\frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial z}
\end{pmatrix},
\]

\[
\frac{\partial^2 f}{\partial u \partial v} = \begin{pmatrix}
\frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\
\frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\
\frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2}
\end{pmatrix}
\]

Coordinate Systems: Derivatives

Gradients transform using the transpose of the inverse matrix (because the coordinate is in the denominator):

\[
g_u = O^{-T} g_x \]
\[
g_x = F^{-T} g_u
\]

Curvatures:

\[
C_u = O^{-T} C_x O
\]
\[
C_x = F^{-T} C_u F
\]

\[
C_{u,kl} = \Sigma_i \Sigma_j O_{ik} C_{x,ij} O_{jl}, \quad C_{x,kl} = \Sigma_i \Sigma_j F_{ik} C_{u,ij} F_{jl}
\]
Coordinate Systems

Summary:

• In crystallographic calculations, we need to use a range of coordinate systems:
  – real and reciprocal space
  – orthogonal, fractional, and grid.

• We also use operators in each space.
  – rotations, translations, and RT.

• Coordinates and operators are related by orthogonalising and fractionalising matrices and their transposes in various combinations.