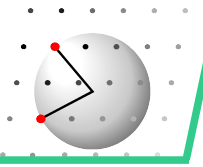
Matrices and Symmetry

Reinhard B. Neder

Lehrstuhl für Kristallographie und Strukturphysik
Department Physik
Universität Erlangen

reinhard.neder@fau.de

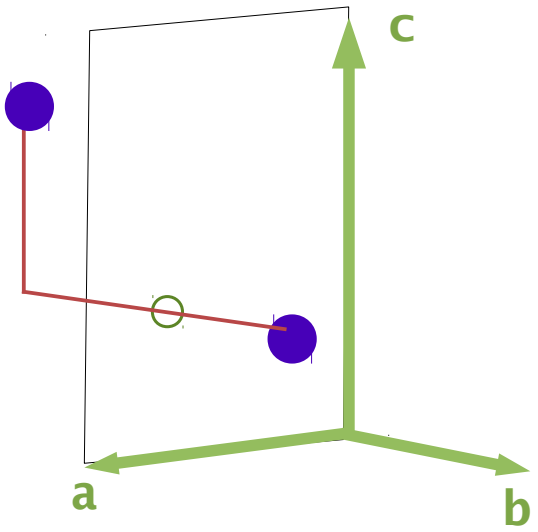
Computational aspects



Monoclinic Space

c-glide plane on x0z plane

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1/2 \end{pmatrix}$$



Combination of mirror matrix plus translation parallel c

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \begin{pmatrix} x \\ -y \\ z + 1/2 \end{pmatrix}$$

Determinant is -1

Trace = +1

Consecutive application of two(any) symmetry operations

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \left[\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \right]$$

$$\vec{x}'' = \underline{\underline{B}} \cdot \vec{x}' + \vec{u} \quad \longleftarrow \quad \vec{x}' = \underline{\underline{A}} \cdot \vec{x} + \vec{t}$$

$$\vec{x}'' = \underline{\underline{C}} \cdot \vec{x} + \vec{w}$$

with: $\underline{\underline{C}} = \underline{\underline{B}} \cdot \underline{\underline{A}}$ and: $\vec{w} = \underline{\underline{B}} \cdot \vec{t} + \vec{u}$

$\{\underline{\underline{A}}, \vec{t}\}$

$\{\underline{\underline{W}}, \vec{w}\}$

Notation in Int. Tables

Augmented form

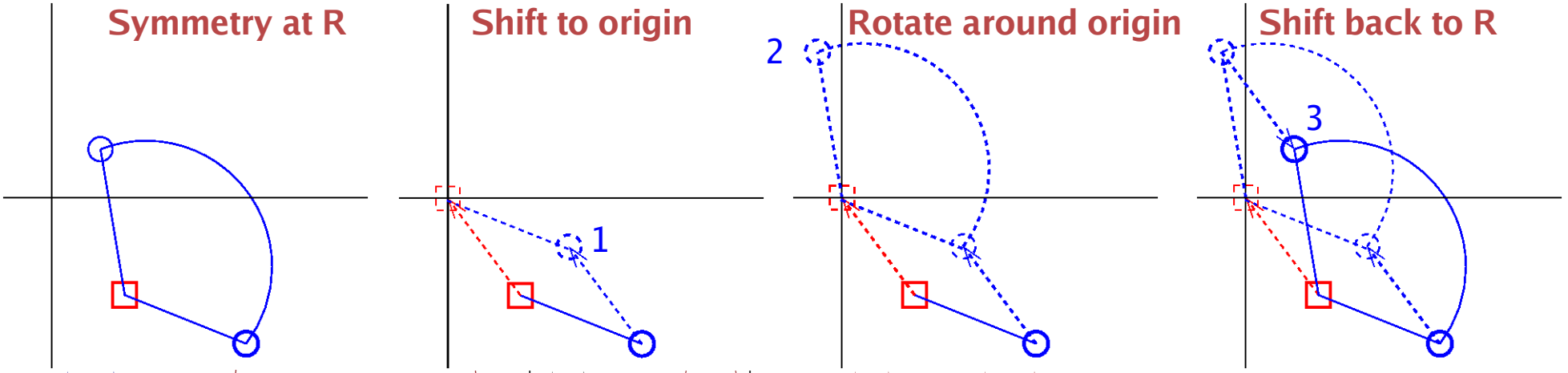
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & t_1 \\ A_{21} & A_{22} & A_{23} & t_2 \\ A_{31} & A_{32} & A_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$\tilde{\vec{x}}'' = \underline{\underline{\tilde{B}}} \cdot \tilde{\vec{x}} \quad \longleftarrow \quad \tilde{\vec{x}}' = \underline{\underline{\tilde{A}}} \cdot \tilde{\vec{x}}$$

$$\tilde{\vec{x}}'' = \underline{\underline{\tilde{C}}} \cdot \tilde{\vec{x}}$$

then: $\underline{\underline{\tilde{C}}} = \underline{\underline{\tilde{B}}} \cdot \underline{\underline{\tilde{A}}}$

Symmetry Operation off the Origin



$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} * \left[\begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix} \right] + \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} + \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} + \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \right] \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix}$$

$$\vec{x}' = \underline{\underline{A}} \cdot \left[\vec{x} - \vec{R} \right] + \vec{t} + \vec{R}$$

$$\vec{x}' = \underline{\underline{A}} \cdot \vec{x} + \vec{t} + \left[\underline{\underline{I}} - \underline{\underline{A}} \right] \cdot \vec{R}$$ Additional translation, independent of x

$$\vec{x}' = \underline{\underline{A}} \cdot \vec{x} + \vec{t}'$$ Requires good documentation

Symmetry operation in reciprocal space

Augmented form

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & t_1 \\ A_{21} & A_{22} & A_{23} & t_2 \\ A_{31} & A_{32} & A_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$\vec{x}'' = \underline{\underline{\tilde{B}}} \cdot \vec{x} \quad \longleftarrow \quad \vec{x}' = \underline{\underline{\tilde{A}}} \cdot \vec{x}$$

$$\vec{x}'' = \underline{\underline{\tilde{C}}} \cdot \vec{x}$$

then: $\underline{\underline{\tilde{C}}} = \underline{\underline{\tilde{B}}} \cdot \underline{\underline{\tilde{A}}}$

$\{\underline{\underline{A}}, \vec{t}\}$

$\{\underline{\underline{W}}, \vec{w}\}$

Notation in Int. Tables

$$(h', k', l', 0) = (h, k, l, 0) * \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} & \tilde{A}_{13} & \tilde{t}_1 \\ \tilde{A}_{21} & \tilde{A}_{22} & \tilde{A}_{23} & \tilde{t}_2 \\ \tilde{A}_{31} & \tilde{A}_{32} & \tilde{A}_{33} & \tilde{t}_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Phase shift: $e^{-2\pi i \vec{h} \vec{T}}$

$$\underline{\underline{\tilde{A}}}' = \underline{\underline{\tilde{G}}} \cdot \underline{\underline{\tilde{A}}} \cdot \underline{\underline{G}}^*$$

→ See presentation R. Cooper

With **G** and **G*** metric and reciprocal metric tensor
Most matrices are identical; exception 3 and 6 in hexagonal metric

Space group C 2/c no 15

Origin at $\bar{1}$ on glide c

Symmetry operations

For (0,0,0)+ set Normal symmetry operations

(1) 1 (2) 2 0, y, ¼ (3) $\bar{1}$ 0, 0, 0 (4) c x, 0, z

For (½, ½, 0)+ set symmetry operations that include the C-centering

(1) t(½, ½, 0) (2) 2(0, ½, 0) ¼, y, ¼ (3) $\bar{1}$ ¼, ¼, 0 (4) n(½, 0, ½) x, ¼, z

Names: (No) type (translation component) location

if present, and NOT Obvious by name

describes points that are on the rotation axis or mirror plane

For $\bar{N}$: (No) $\bar{N}$ location; location

In Fm$\bar{3}$m (41) $\bar{4}$- x,0,0; 0,0,0

axis; inversion center

Generators (1); t(1,0,0); t(0,1,0); t(0,0,1); t(½, ½, 0); (2); (3)

Positions Coordinates Subset to generate all symmetry matrices

Multiplicity
Wyckoff letter
Site symmetry (0,0,0)+ (½ , ½ , 0)+ Add these vectors to ALL atom positions listed

8 f 1 (1) x, y, z (2) $\bar{x}$, y, $\bar{z}$ + ½ (3) $\bar{x}$, $\bar{y}$, $\bar{z}$ (4) x, $\bar{y}$, z + ½

8 = No of atom positions

(2) $\bar{x}$, y, $\bar{z}$ + ½ (Number) identical to list on previous page

$$\begin{pmatrix} -x \\ y \\ -z+1/2 \end{pmatrix}_2 \xleftarrow{(2)} \begin{pmatrix} x \\ y \\ z \end{pmatrix}_1$$

$$\begin{pmatrix} -x \\ y \\ -z+1/2 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1/2 \end{pmatrix}$$

Result of Sym. Op no (2) You always start with atom no. (1) x, y, z

-x: The linear function no 2 of (x,y,z) always results in -x

-x = -1 x + 0 y + 0 z First row of symmetry matrix

Not a scew axis

Translation NOT parallel to rotation axis

(2) 2 0, y, ¼

Determinant, Trace of W gives type
Solution to $\vec{r} = \underline{W}\,\vec{r}$ Give points on axis/plane

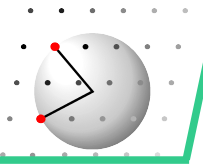
Computational aspects

$$\begin{pmatrix} result_1 \\ result_2 \\ result_3 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} * \begin{pmatrix} source_1 \\ source_2 \\ source_3 \end{pmatrix} + \begin{pmatrix} trans_1 \\ trans_2 \\ trans_3 \end{pmatrix}$$

$$result_L = \left[\sum_{K=1}^{3} A_{L\,K} \cdot source_K \right] + trans_L$$

```
for all L in 1 to 3 DO
    result( L ) = 0
    for all K in 1 to 3 DO
        result( L ) = result( L ) + A( L,K ) * source( K )
    result( L ) = result( L ) + trans( L )
```

```
for all L in 1 to 3 DO
    result( L ) = trans( L )          ! less obvious, faster
    for all K in 1 to 3 DO
        result( L ) = result( L ) + A(L,K) * source( K )
```



Augmented form

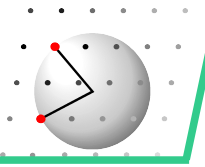
$$\begin{pmatrix} result_1 \\ result_2 \\ result_3 \\ 1 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & 1 \\ A_{21} & A_{22} & A_{23} & 1 \\ A_{31} & A_{32} & A_{33} & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} source_1 \\ source_2 \\ source_3 \\ 1 \end{pmatrix}$$

$$result_L = \left[\sum_{K=1}^{4} A_{L\,K} \cdot source_K \right]$$

```
for all L in 1 to 4 DO
    result( L ) =  0
    for all K in 1 to 4 DO
        result( L ) =  result( L )  +  A( L,K ) * source( K )
```

```
for all L in 1 to 3 DO
    result( L ) =  0
    for all K in 1 to 4 DO
        result( L ) =  result( L )  +  A( L,K ) * source( K )
result( 4 ) =  1
```

Do not bother unless you do this 10^9 times



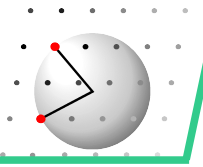
Sample Codes

FORTRAN 77 or older

```
      real matrix(3,3),result(3),source(3),trans(3)
. . .
      matrix(1,1) = 1.000
. . .
      do 10 i=1,3
      result(i) = trans(i)
      do 10 k=1,3
      result(i)=result(i)+matrix(i,k)*source(k)
10    continue
```

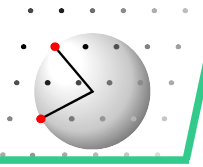
Fixed line format

- Statements in columns 7 to 72
- Labels in columns 1 to 5
- Continuation flag in column 6



fortran2008

```
IMPLICIT NONE
REAL, DIMENSION(3,3) :: matrix = 0.0 ! Symmetry matrix
REAL, DIMENSION(3)   :: trans  = 0.0 ! translational part
REAL, DIMENSION(3)   :: result = 0.0 ! Coordinates of result
REAL, DIMENSION(3)   :: source = 0.0 ! Coordinates of input atom
. . .
matrix(1,1) = 1.000
. . .
DO i = 1, 3                          ! loop over result indices
   result(i) = trans(i)              ! initialize with translational part
   DO k = 1, 3                       ! loop over matrix column index
      result(i) = result(i) + matrix(i,k) * source(k)
   END DO                            ! end of loop over matrix
END DO                               ! end of loop over result indices
```

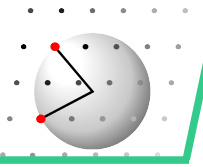


Sample Codes

fortran2008

```
IMPLICIT NONE
REAL, DIMENSION(3,3) :: matrix = 0.0 ! Symmetry matrix
REAL, DIMENSION(3)   :: trans  = 0.0 ! translational part
REAL, DIMENSION(3)   :: result = 0.0 ! Coordinates of result
REAL, DIMENSION(3)   :: source = 0.0 ! Coordinates of input atom
. . .
matrix(1,1) = 1.000
. . .
loopr: DO i = 1, 3                   ! loop over result indices
   result(i) = trans(i)              ! initialize with translational part
   loopm: DO k = 1, 3                ! loop over matrix column index
      result(i) = result(i) + matrix(i,k) * source(k)
   END DO loopm                      ! end of loop over matrix
END DO loopr                         ! end of loop over result indices
```

Indentation + labels keep code more legible

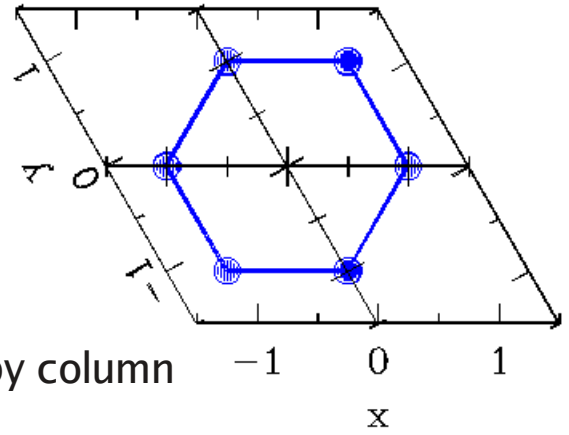


fortran2008

```
IMPLICIT NONE
REAL, DIMENSION(3,3) :: matrix = RESHAPE((/1.,1.,0., -1.,0.,0., 0.,0.,1./),SHAPE(matrix))
REAL, DIMENSION(3)   :: trans  = (/0., 0., 0./) ! translational part
REAL, DIMENSION(3)   :: result = 0.             ! Coordinates of result
REAL, DIMENSION(3)   :: source = (/1., 0., 0./) ! Coordinates of input atom

result = MATMUL(matrix, source) + trans ! use intrinsic function MATMUL
```

$$\begin{pmatrix} 1 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$



$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \begin{pmatrix} x-y \\ x \\ z \end{pmatrix}$$

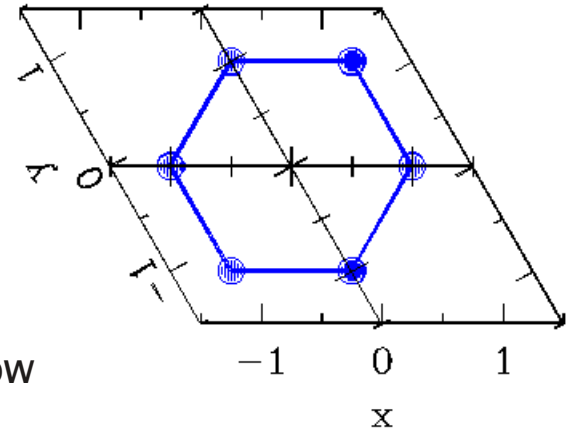
fortran initialises column by column

Sample Codes

C; C++

```
float matrix[3][3] = {1.0,-1.0, 0.0,  1.0, 0.0, 0.0,  0.0, 0.0, 1.0};
float trans[3]     = {0.0, 0.0, 0.0};
float result[3]    = {0.0, 0.0, 0.0};
float source[3]    = {1.0, 0.0, 0.0};
…
for ( i = 0 ; i < 3 ; i++ )
  {
   result[i] = trans[i];
   for ( k = 0 ; k < 3 ; k++ )
     {
        result[i] = result[i] + matrix[i][k]*source[k];
     }
  }
```

$$\begin{pmatrix} 1 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$



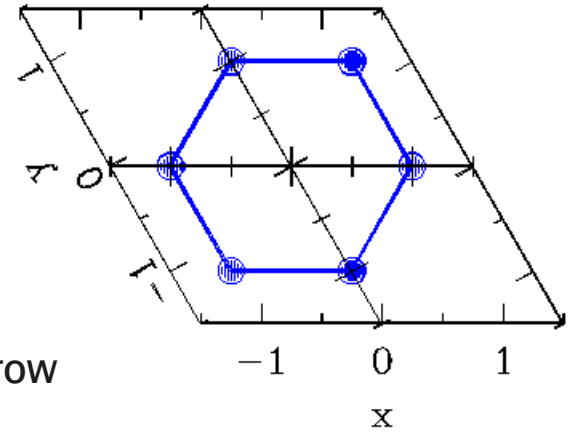
$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \begin{pmatrix} x-y \\ x \\ z \end{pmatrix}$$

c c++ initialises row by row

Python with numpy

```
import numpy as np
matrix = np.array( ((1.0,-1.0, 0.0), (1.0, 0.0, 0.0), ( 0.0, 0.0, 1.0)))
trans  = np.array( ((0.0, 0.0, 0.0)))
source = np.array( ((1.0, 0.0, 0.0)))
result = np.dot(matrix,source) + trans
```

$$\begin{pmatrix} 1 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$



$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \begin{pmatrix} x-y \\ x \\ z \end{pmatrix}$$

python initialises row by row

Many more than one atom — Typically up to 1000, sometimes 1 Million — **Be flexible!**

Many more than one matrix — **How many do you need?**

Fm$\bar{3}$m has 192 symmetrically equivalent atoms
To generate all you need just 7 matrices !

Even 192 matrices is a small number make fixed arrays

Add one rank for the number of matrices

Add one rank for the number of atoms

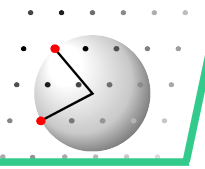
```
IMPLICIT NONE
REAL, DIMENSION(NMATRIX,4,4) :: matrix = 0.0  ! REAL, DIMENSION(4,4,NMATRIX) :: matrix

REAL, DIMENSION(NATOMS,4)    :: result = 0.0
REAL, DIMENSION(NATOMS,4)    :: source = 0.0
. . .
result = MATMUL(matrix(1,:,:), source(1,:) )
```

Better suited to storage sequence of fortran

Tell program, which (4,4) matrix to use

Equivalent approaches with structured variables, „classes“, and overridden operators

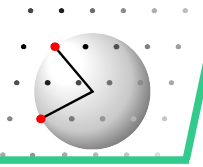


Example 1: Derive all symmetry matrices of a space group

Input: Identity matrix (1) 1

List of all generators **Subset of all symmetry matrices; cyclically applied give all matrices**

Conveniently as (4x4) matrix



Generators (1); t(1,0,0); t(0,1,0); t(0,0,1); t(½, ½, 0); (2); (3)

Positions Coordinates Subset to generate all symmetry matrices

Multiplicity
Wyckoff letter
Site symmetry (0,0,0)+ (½ , ½ , 0)+ Add these vectors to ALL atom positions listed

8 f 1 (1) x, y, z (2) $\bar{x}$, y, $\bar{z}$ + ½ (3) $\bar{x}$, $\bar{y}$, $\bar{z}$ (4) x, $\bar{y}$, z + ½

8 = No of atom positions

(2) $\bar{x}$, y, $\bar{z}$ + ½ (Number) identical to list on previous page

$$\begin{pmatrix} -x \\ y \\ -z+1/2 \end{pmatrix}_2 \xleftarrow{(2)} \begin{pmatrix} x \\ y \\ z \end{pmatrix}_1$$

$$\begin{pmatrix} -x \\ y \\ -z+1/2 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1/2 \end{pmatrix}$$

Result of Sym. Op no (2) You always start with atom no. (1) x, y, z

-x: The linear function no 2 of (x,y,z) always results in -x

-x = -1 x + 0 y + 0 z First row of symmetry matrix

Not a scew axis

Translation NOT parallel to rotation axis

(2) 2 0, y, ¼

Determinant, Trace of W gives type
Solution to $\vec{r} = \underline{W}\,\vec{r}$ Give points on axis/plane

Input: Identity matrix (1) 1

List of all generators **Subset of all symmetry matrices; cyclically applied give all matrices**

Conveniently as (4x4) matrix

In: C2/c

(2) 2 0, z, ¼
(3) $\bar{1}$ 0,0,0
t(½, ½, 0)

$\underline{\mathbf{S}}_1 = \underline{\mathbf{I}}$ create first symmetry matrix as unit matrix $\underline{\mathbf{I}}$

For all generators $\underline{\mathbf{G}}$

For all old symmetry matrices $\underline{\mathbf{S}}_{old}$

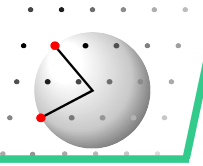
$\underline{\mathbf{S}}_{new} = \underline{\mathbf{G}}\,\underline{\mathbf{S}}_{old}$ create new symmetry matrix

Update number of old symmetry matrices

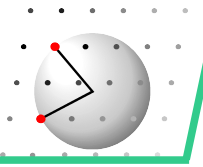
Beware of 3-fold rotations !

They triple number of Symmetry operations

All others double.



Sample Source Code
See separate listing



Example 2

Derive all atom positions in a unit cell **Purpose ?**

- calculate structure factor

  one unit cell sufficient

  speed, as repeated at every refinement cycle

  → ignore special positions
  create as many copies as on general position; *Occupancy*
  ignore centering
  ignore inversion

- visualize crystal structure

  one to a few unit cells usually sufficient

  no multiple atoms ==> ! special positions!

- calculate (physical, chemical, biochemical, thermodynamic, …,) properties

  one unit cell sufficient

  no multiple atoms ==> ! special positions!

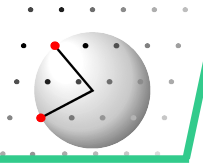
- create multiple unit cells for later modification

  MANY unit cells

  no multiple atoms ==> ! special positions!

  speed for later modifications / calculations

  memory



Derive all atom positions in a unit cell

Input: Atom positions (x,y,z)

List of all generators **Subset of all symmetry matrices; cyclically applied give all matrices**

Conveniently as (4x4) matrix

In: C2/c

(2) 2 0, z, ¼
(3) $\bar{1}$ 0,0,0

For all generators **G**

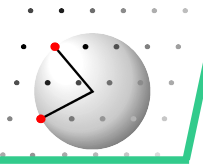
For all old atom positions (x,y,z)

$\vec{x}_{new} = \underline{\underline{G}}\,\vec{x}_{old}$ create new atom position

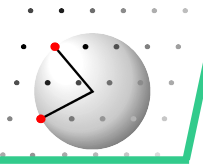
Normalize x,y,z to 0 <= x,y,z < 1

IF equal to any previous atom

Skip atom, move to next generator

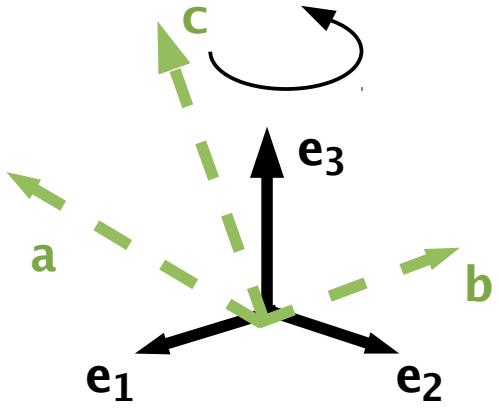


Sample Source Code
See separate listing



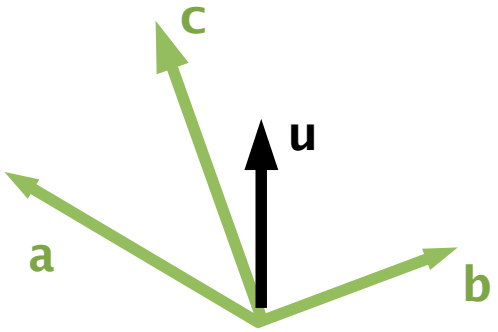
Example 3: General symmetry operation in triclinic space

Rotation around e_3 in cartesian



$$\begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Rotation around arbitrary axis in crystal



Transform cartesian axes to crystal axes such that u || e_3

Solution:
D.E. Sands
Vectors and Tensors in Crystallography

$$W_{ij} = u_i u_j^* + (\delta_{ij} - u_i u_j^*)\cos(\alpha) + g_{ik}\,\epsilon_{klj}\,u_l \sin(\alpha)$$

$$\vec{w}\,' = \vec{w} + \left[\underline{\underline{I}} - \underline{\underline{W}}\right]\cdot\vec{R}$$

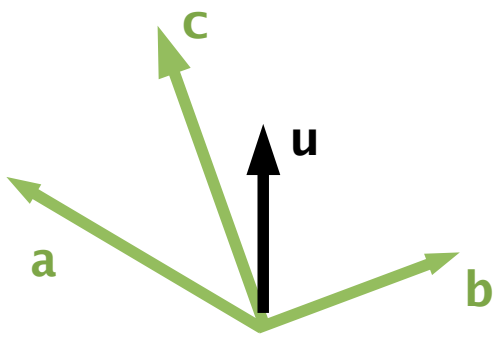
See following presentation
By R. Cooper

Example 3: General symmetry operation in triclinic space

Rotation around arbitrary axis in crystal

$$W_{ij} = u_i u_j^* + (\delta_{ij} - u_i u_j^*)\cos(\alpha) + g_{ik}\,\epsilon_{klj}\,u_l \sin(\alpha)$$

$$\vec{w}\,' = \vec{w} + \left[\underline{\underline{I}} - \underline{\underline{W}}\right]\cdot\vec{R}$$



u_i	Elements of u with respect to crystal axes
u_j^*	Elements of u with respect to reciprocal crystal axes
δ_{ij}	Kronecker symbol
α	Rotation angle
g_{ik}	Elements of metric tensor
ϵ_{klj}	Elements of permutation tensor

+1 for even permutations

-1 for odd permutations

0 if any two indices are equal

Solution:
D.E. Sands
Vectors and Tensors in Crystallography

Homework

Write a program that:

Rotates an atom by 90° around the c-axis

Modify

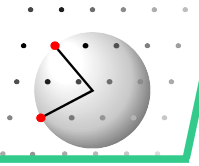
Rotates an atom by 90° around an axis parallel to the c-axis at (¼, ¼, 0)

Modify

Rotates an atom by 90° around an axis parallel to the c-axis at (¼, ¼, 0)

Create all copies of original atom

Pay attention to special positions!



References

D.E. Sands
Vectors and Tensors in Crystallography

International Tables for Crystallography Vol A

Giacovazzo et al.
Fundamentals of Crystallography