

ACQua - Documentation

Max Marcus

July 19, 2012

The Documentation shall contain all informations regarding the program ACQua. Full descriptions of routines, modules and functions as well as the theoretical background.

The subroutines shall be described in the section of the module they belong to and be in alphabetical order therein. Variables shall be described and it shall be stated at which point they are used again.

A list of user keywords shall also be herein.

Each subroutine documentation shall start with a box containing informations about: variables, subroutines calling and called subroutines. The variables shall be stated with name, followed by a two or three digit code in brackets specifying the variable type (Integer I, Real*8 R8, Character(LEN=20) C20, etc.) and the source and fate of it (i: inherited, g: generated, n: never to be used againm, a: altered).

Contents

| | | |
|----------|--------------------------------|----------|
| 1 | Modules and Subroutines | 1 |
| 1.1 | pointgroups.f90 | 1 |
| 1.1.1 | c2vz | 2 |
| 1.1.2 | c2x | 3 |
| 1.1.3 | ci | 4 |

1 Modules and Subroutines

1.1 pointgroups.f90

Objective If a molecule is calculated, the module *pointgroups.f90* contains all necessary subroutines to generate equivalent atoms from the atoms given in the input. It takes the position matrix (POSMAT) and appends equivalent atoms to give the full set of atoms generating the molecule specified in the input.

Theory Let $\{\mathbf{G}_i\}_{i=1}^g$ be a full representation of the pointgroup G and g be its order (i.e. the number of different \mathbf{G} in $\{\mathbf{G}\}$).

The symmetry operation G_i can be represented by a 3×3 matrix \mathbf{G}_i in Cartesian coordinates.

If $\mathbf{R}^{(0)}$ is the matrix containing the position vectors (here: positions of inequivalent atoms specified in the input), the image of $\mathbf{R}^{(0)}$ under \mathbf{G}_i can be obtained by

$$\mathbf{R}^{(i)} = \mathbf{G}_i \mathbf{R}^{(0)} \quad (1)$$

(in this notation the position vectors are aligned in columns rather than in rows in \mathbf{R} . Hence the transformation vectors in \mathbf{G}_i must be aligned in rows. The purpose will become clear later).

If this is done for all \mathbf{G}_i in $\{\mathbf{G}_i\}_{i=1}^g$ a set of g position matrices is obtained $\{\mathbf{R}^{(i)}\}_{i=1}^g$ (note: $\mathbf{R}^{(0)}$ is equal to the matrix obtained for the identity transformation being the trivial symmetry operation).

The obtained position vectors $\{\mathbf{r}_k\}_{k=1}^{n \times g}$ (n being the number of inequivalent atoms) do not need to be all

unique. The so called *unique set* of the position vectors is then $\{\mathbf{r}_k^{\text{uni.}}\}_{k=1}^m$.

The matrices representing the symmetry operations can be obtained from the point group tables. The entry in the matrix is equivalent to the character of the functions x, y, z under the according symmetry operation.

Implementation Each point group is assigned a number according to the International tables of Crystallography (ITA). This number is given in the input and the corresponding subroutine is called. The position matrix (POSMAT), read from the input, is given as input. This matrix has the form

$$\mathbf{R}^{(0)} = \begin{pmatrix} 1 & 2 & \cdots & n \\ E_1 & E_2 & \cdots & E_n \\ x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \\ z_1 & z_2 & \cdots & z_n \end{pmatrix}, \quad (2)$$

where the first row is an index from 1 to n , the second row is the corresponding element number and the third to fifth row is the position vector. n is stored as NUMINAT in the code. POSMAT thus has the dimensions $5 \times \text{NUMINAT}$.

This matrix is handed over to the subroutines containing the symmetry operation matrices $\{\mathbf{G}_i\}_{i=1}^g$ of G .

The matrices are multiplied using *matmult/matmath.f90* and each $\mathbf{R}^{(i)}$ is stored in different arrays. They are compared in *matcomp/matmath.f90* in pairs resulting ultimately in the POSMAT of dimensions $5 \times \text{NUMAT}$ (NUMAT being m above).

The transformation matrices are constructed in the subroutines themselves. The actual transformation matrix is given in the block $3-5 \times 3-5$ in the matrix, whereas 11 and 22 are elements of value 1 while all others are 0. The block $1-2 \times 1-2$ defines a metric to conserve element and generating atom information to be used in the SCF.

1.1.1 c2vz

NUMINAT (Ii): number of inequivalent atoms, specified in input, inherited from *pointgroup/symmetry.f90*

NUMAT (Ig): total number of atoms, obtained after comparing of $\mathbf{R}^{(i)}$

SMAT1-3 (In): number of columns in MAT1-3

C2Z(5,5) (R8n): matrix of $\mathbf{C}_2^{(z)}$

SXZ(5,5) (R8n): matrix of $\sigma^{(xz)}$

SYZ(5,5) (R8n): matrix of $\sigma^{(yz)}$

C2ZPOS(:, :) (R8n): matrix containing created atoms by C2Z

SXZPOS(:, :) (R8n): matrix containing created atoms by SXZ

SYZPOS(:, :) (R8n): matrix containing created atoms by SYZ

MAT1(:, :) (R8n): matrix containing inequivalent atoms from POSMAT and C2ZPOS

MAT2(:, :) (R8n): matrix containing inequivalent atoms from POSMAT and SXZPOS

MAT3(:, :) (R8n): matrix containing inequivalent atoms from POSMAT and SYZPOS

POSMAT(:, :) (Ia): matrix containing position vectors of nuclei.

Called by: *pointgroup/symmetry.f90*

Calls: *matmult/matmath.f90*, *matcomp/matmath.f90*

c2vz contains the information for the point group $C_{2v}^{(z)}$ with the rotation axis being aligned to the z -axis. It's ITA number is 15.

Table 1: Point group table for $C_2(x)$.

| | E | $C_2^{(z)}$ | $\sigma_v^{(xz)}$ | $\sigma_v^{(yz)}$ | $g = 2$ |
|-------|-----|-------------|-------------------|-------------------|--------------------|
| A_1 | 1 | 1 | 1 | 1 | z, x^2, y^2, z^2 |
| A_2 | 1 | 1 | -1 | -1 | R_z, xy |
| B_1 | 1 | -1 | 1 | -1 | x, R_y, xz |
| B_2 | 1 | -1 | -1 | 1 | y, R_x, yz |

The transformation matrices C2Z(5,5), SXZ(5,5) and SYZ(5,5) are then given by

$$\mathbf{C2Z} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{SXZ} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} \quad \mathbf{SYZ} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

The resulting $\mathbf{R}^{(1)}$, $\mathbf{R}^{(2)}$, $\mathbf{R}^{(3)}$ (obtained via *matmult/matmath.f90* of POSMAT and C2Z, POSMAT and SXZ, POSMAT SYZ) are stored in C2ZPOS(5,NUMINAT), SXZPOS(5,NUMINAT) and SYZPOS(5,NUMINAT). C2ZPOS is then compared to POSMAT (via *matcomp/matmath.f90*) and the resulting matrix is stored in MAT1(5,SMAT1), which is compared zu SXZPOS and the resulting matrix (MAT2(5,SMAT2)) is compared to SYZPOS. The generated matrix MAT3(5,SMAT3) is the matrix containing all atoms building up the molecule and so it is renamed as POSMAT(5,NUMAT) and NUMAT = SMAT3.

POSMAT is then printed and MAT1, MAT2 and MAT3 as well as C2XPOS, SXZPOS, SYZPOS and C2Z, SXZ, SYZ are deallocated.

1.1.2 c2x

| |
|---|
| NUMINAT (Ii): number of inequivalent atoms, specified in input, inherited from <i>pointgroup/symmetry.f90</i> |
| NUMAT (Ig): total number of atoms, obtained after comparing of $\mathbf{R}^{(1)}$ |
| SMAT1 (In): number of columns in MAT1 |
| C2(5,5) (R8n): matrix of $\mathbf{C}_2^{(x)}$ |
| C2XPOS(:, :) (R8n): matrix containig created atoms by C2 |
| MAT1(:, :) (R8n): matrix containing inequivalent atoms from POSMAT and C2XPOS |
| POSMAT(:, :) (Ia): matrix containing position vectors of nuclei. |
| Called by: <i>pointgroup/symmetry.f90</i> |
| Calls: <i>matmult/matmath.f90</i> , <i>matcomp/matmath.f90</i> |

c2x contains the information for the point group C_2 with the rotation axis being aligned to the x -axis. It's ITA number is 3.

Table 2: Point group table for $C_2(x)$.

| | E | $C_2^{(x)}$ | $g = 2$ |
|-----|-----|-------------|-----------------------------|
| A | 1 | 1 | $x, R_x, x^2, y^2, z^2, yz$ |
| B | 1 | -1 | $y, z, R_{y,z}, xy, xz$ |

The transformation matrix C2(5,5) is then given by

$$\mathbf{C2} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

The resulting $\mathbf{R}^{(1)}$ (obtained via *matmult/matmath.f90* of POSMAT and C2) is stored in C2XPOS(5,NUMINAT) and then compared to POSMAT (via *matcomp/matmath.f90*). The resulting matrix is stored in MAT1(5,SMAT1). Since there is only one non-trivial symmetry operation, MAT1 is the matrix containing all atoms building up the molecule and so it is renamed as POSMAT(5,NUMAT) and NUMAT = SMAT1. POSMAT is then printed and MAT1 as well as C2XPOS and C2 are deallocated.

1.1.3 ci

NUMINAT (Ii): number of inequivalent atoms, specified in input, inherited from *pointgroup/symmetry.f90*
 NUMAT (Ig): total number of atoms, obtained after comparing of $\mathbf{R}^{(i)}$
 SMAT1 (In): number of columns in MAT1
 CIM(5,5) (R8n): matrix of \mathbf{i}
 CIPOS(:, :) (R8n): matrix containing created atoms by CIM
 MAT1(:, :) (R8n): matrix containing inequivalent atoms from POSMAT and CIPOS
 POSMAT(:, :) (Ia): matrix containing position vectors of nuclei.

Called by: *pointgroup/symmetry.f90*

Calls: *matmult/matmath.f90*, *matcomp/matmath.f90*

ci contains the information for the point group C_i . The corresponding ITA number is 2.

Table 3: Point group table for C_i .

| | \mathbf{E} | \mathbf{i} | $g = 2$ |
|----------------|--------------|--------------|--------------------------------------|
| \mathbf{A}_g | 1 | 1 | $R_{x-z}, x^2, y^2, z^2, xy, xz, yz$ |
| \mathbf{A}_u | 1 | -1 | x, y, z |

The symmetry operation matrix constructed is CIM(5,5) and has the form

$$\mathbf{CIM} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

The resulting $\mathbf{R}^{(1)}$ (obtained via *matmult/matmath.f90* of POSMAT and CIM) is stored in CIPOS(5,NUMINAT) and then compared to POSMAT (via *matcomp/matmath.f90*). The resulting matrix is stored in MAT1(5,SMAT1). Since there is only one non-trivial symmetry operation, MAT1 is the matrix containing all atoms building up the molecule and so it is renamed as POSMAT(5,NUMAT) and NUMAT = SMAT1. POSMAT is then printed and MAT1 as well as CIPOS and CIM are deallocated.