Kusevska Elena

Universidad Autonoma de Madrid

Report of symmetry analyser program

In this report I will describe the basis on which the algorithm determines molecular symmetry and how to use the program.

1. Description of the algorithm

The program is distributed in two source code files. The file called program.f95, contains the main body of the program and two very simple subroutines, the first for assigning values to arrays containing information about atomic labels and atomic masses for all atoms of the periodic table, and the second for determining the computational time spent by the program. The file called modsym.f95 is a module that contains all the subroutines used for applying the symmetry operations on the molecule, and determining if the molecule contains these symmetry elements.

First, the program reads the input geometry of the molecule. If the molecule is a diatomic, it immediately determines wether it is C ∞v or D ∞h. If both atoms are the same, it must be D ∞h, if they are different it must be C ∞v. And it stops.

If the molecule is not a diatomic, this step is ignored and the program proceeds normally, assigning the atomic masses and determining the coordinates of the centre of mass using the formula:

$$\text{center}_{\text{x-coordinate}} = \frac{\sum_i m_i x_i}{\sum_i m_i} \qquad \text{center}_{\text{y-coordinate}} = \frac{\sum_i m_i y_i}{\sum_i m_i} \qquad \text{center}_{\text{z-coordinate}} = \frac{\sum_i m_i z_i}{\sum_i m_i}$$

Then it moves the centre if mass of the molecule to the origin of the Cartesian coordinate system. This is very useful because the centre of mass is the one point that is not moved by any symmetry operation, so all symmetry elements pass through it. So, by placing the centre of mass in the origin, the program is making sure that every symmetry element passes through the origin.

Then it determines the elements of the matrix of the inertia tensor, and it diagonalizes the inertia tensor using the DSYEV lapack subroutine. On input, the DSYEV subroutine takes the matrix inert, which represents the inertia tensor. On output it gives the same matrix (inert), but now it contains the eigenvectors of the inertia tensor in its columns. The eigenvalues give the moduli of the three principal moments of inertia and the corresponding eigenvectors are the directions of the moments of inertia.

When running this subroutine, the program gives the value of 'info' in the output. If info = 0, the diagonalization completed successfully.

Then the program checks to see if the molecule belongs to one of the cubic symmetry groups. If all three moments of inertia have roughly the same value, then the molecule is cubic. If the molecule is not cubic, then the program checks if the molecule is linear. If one of the

eigenvalues of the inertia tensor is zero or very close to zero, then one of the three principal moments of inertia is zero, and so the molecule is linear. If the molecule has an inversion centre, then it must belong to the D ∞h point group, if not to the C ∞v point group.

If the molecule is neither linear, nor cubic, then the program places all the coordinates and atomic masses in one matrix, where each column vector represents one atom. Through the use of matrix algebra on this matrix, the program can apply symmetry operations and check if these operations are present in the molecule.

Then the program searches for the maximum order of rotation around each of the moment of inertia eigenvectors. The one with the highest maximum order of rotation gives the direction of the main axis of rotation for the molecule.

If the maximum order of the rotation is 1, then the group does not have rotational symmetry, so the program searches for one of the lower symmetry, noncyclic point groups. First, it searches for a horizontal plane normal to one of the eigenvectors of the inertia tensor, and if it finds one, it assigns the Cs point group to the molecule. If not, then it checks for an inversion, and if it finds one, it assigns the Ci point group to the molecule. If it does not find an inversion centre, then the molecule must belong to the C1 point group.

In case there is rotational symmetry, the program determines the main axis of rotation to be along the eigenvector for which the maximum order of rotation was found. It then aligns the molecule along this eigenvector for the rest of the search.

Then it has to check if the point group is dihedral. For this it needs to find a secondary axis of rotation. In order to know where in space to search for this axis, I have chosen the following strategy: First, the program builds the distance matrix. Then, it searches for any two symmetrically equivalent atoms, whose midpoint does not lie in the centre of mass of the molecule, and that either they, or their midpoint lies in the xy plane, when the molecule is positioned with the main axis of rotation along the z – axis of the coordinate system. This is shown for allene and cyclohexane in the “chair” - conformation in Figure1 and Figure2.

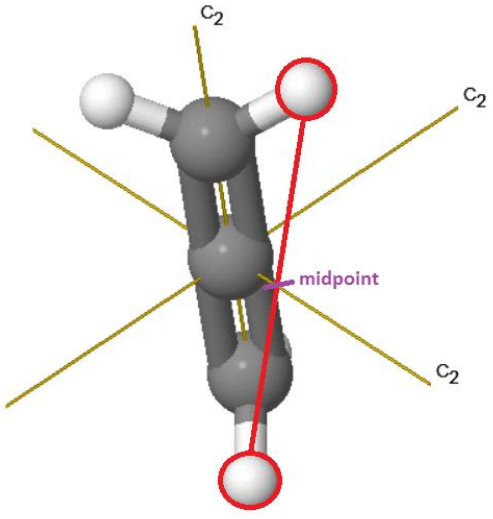

Figure 1 allene

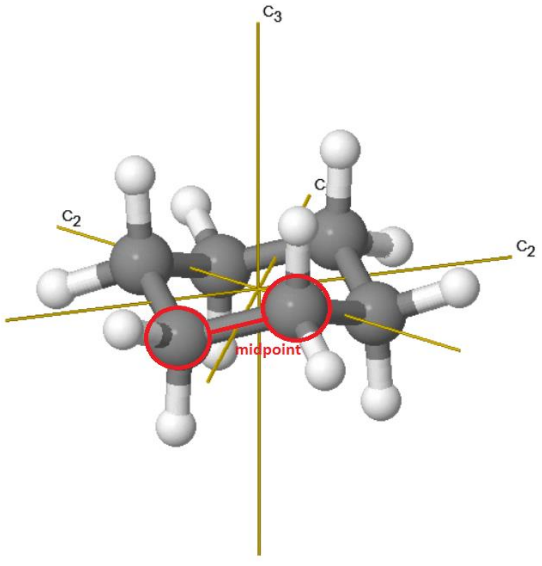

Figure 2 benzen - “chair” conformation

I didn’t find any source on how to determine the position of the secondary axis of rotation, I was just looking at a lot of different molecules with dihedral symmetry and I realized that the secondary axis always passes either through an atom lying in the xy plane or between two atoms, where the midpoint lies in the xy plane (since it must be normal to the z-axis).

If the program has not managed to find two such atoms, then the point group must be cyclic. First the program searches for a horizontal reflection plane. If it finds this, the point group of the molecule is Cnh. If it doesn’t find a horizontal reflection plane, then it searches for a vertical reflection plane along one of the two symmetrically equivalent atoms that did not fulfil the requirement that they or their midpoint lye in the xy-plane. If the molecule does not have a vertical plane of symmetry, then the program checks if it belongs to one of the Sk (k = 2n) point groups. Finally, if the molecule does not have an improper rotation axis, it must belong to the Cn point groups.

If the program has managed to find SEA such that their coordinates or those of their midpoint can be used to find the secondary rotation axis, then it starts searching through the dihedral groups. First it searches for a secondary rotation axis. If it finds it, then the point group is Dihedral. Then it searches for a horizontal reflection. If it finds a horizontal reflection plane, the point group is Dnh, if not it searches for a vertical reflection plane, which is not parallel to the secondary rotation axis (a dihedral plane). If it finds it, then the point group is Dnd. If it doesn’t find a vertical reflection plane, then the point group of the molecule is Dn.

If it doesn't find a secondary rotation axis, then the group cannot be dihedral, the program searches through the cyclic groups in the same way as was described in the previous paragraph.

2. Deviation from perfect symmetry

I found several papers describing the continuous symmetry measure method, but I did not have time to implement them in the program. So I decided to approach the measure of deviation from perfect symmetry in a simpler way. When the program finds a symmetry element, it normalises the atomic distances in a way that the greatest distance from the centre of mass is equal to 1. This allows different molecular structures to be compared with the same measure

It then calculates the distance between the atomic positions in the molecule before and after applying the symmetry element, for every atomic position. The square of these distances divided by the total number of atoms gives the variance and the square root of the variance, the standard deviation. So, the program provides the standard deviation of some of the symmetry elements as a measure of how close to perfect symmetry the molecule is.

As for the accepted error for a symmetry element, I just placed a simple threshold of 0.01 for all symmetry elements and it worked. I only had to increase it to 0.015 for 1,1-chloroethene. I think for molecules with more atoms or more complex symmetry it would be necessary to define a way to quantify this value, and to have it change depending on the molecule.

3. User’s manual

To run the program, just compile it with the command:

```
gfortran program.f95 modsym.f95 -llapack -lblas
```

or any other standard command including lapack and blas libraries, for the Fortran compiler that you are using. I have only used Fortran90/Fortran95 features in this program.

Create an input file called input with the coordinates of the molecule in xyz format in the same folder as the source code, compile and run.

If you want to run the program for a second time, first remove the output and lwork files, because they are defined as ‘new’ in the program, so if they already exist Fortran gives a work-time error.

The program gives the results in a file called output. In this file it gives the input geometry, the eigenvalues and eigenvectors from the diagonalization of the inertia tensor, the distance matrix, the symmetrically equivalent atoms used in the search for vertical planes and a secondary rotation axis, the symmetry group and the standard deviation for groups with at least a main axis of rotation.

I have only run the program in Linux, but as far as I know, Fortran is not platform specific, so it should run on windows and mac as well.