NAME

coord.in, — Coordinate definition input file for **surfgen**.

SYNOPSIS

This input file is used by coupled potential surface fitting program **surfgen** and its potential evaluation library. For a detailed description of the program and the evaluation library, see `surfgen`(1), `potlib`(1)

Place **coord.in** in the same directory where **surfgen** is being called.

DESCRIPTION

coord.in contains definition of the type, scaling mode, defining atoms and scaling parameters for all the internal coordinates that are used as a basis for the expansion of Hd.
coord.in also sets the maximum order for each coordinate, as well as additional order conditions in the form of linear inequality constraints.

For coordinates that are related by symmetry operations, only one of them should be defined. All the symmetry related coordinates will be automatically added to form a group of coordinates, refered to as *coordinate sets*.
For a list of active symmetry operations, call **surfgen** with *jobtype=-1*.

Developed by Yarkony group, Johns Hopkins University 2010-2013.

INPUT FORMAT

The file consists of the definitions of coordinate sets and the additional order conditions. The entire file use automatic formating, positions and the number of spaces or length of numbers will not affect the input.

Coordinate and Condition Counts

First line of **coord.in** is a comment line left for the user to describe the coordinate system.
The second line consists of two integers, providing the number of coordinate sets **nCoordSets** and the number of additional conditions **nAddCond**.

Coordinate Set Definitions

Coordinate set definition is composed of the definition of each of the **nCoordSets** coordinate sets. For each coordinate, the definition consists of 3 or 4 lines, depending on the type and scaling mode of coordinates.

General Format

The first line is a comment line. The second line consists of three integers, which defines the type, scaling mode and maximum order of the coordinate. When maximum order is set to 0, no maximum will be imposed on this coordinate.
The third line consists of four numbers, *A1*, *A2*, *A3* and *A4*. All or some of these defines the indices of atoms that are involved in the coordinates. For coordinate types that involve less than 4 atoms, the last one or more integers will not be used by the program. However, all 4 numbers will still need to be present, even though some are not used.
The fourth line is present for type-scaling mode combinations that require scaling parameter input. This line contains two floating point or scientific numbers that will be used to define the coordinates. When scaling parameters are not needed, this field should *NOT* be present.

See the following section for details about the available coordinate types and scaling modes and the number of scaling parameters needed for each combination.

Example

This is an example for out-of-plane bend

```
out-of-plane H O C H
-1      0       0
1       5       2       4
1.000 0.000
```

Coordinate Types

The following coordinate types are currently implemented or planned:

0 Distance or scaled distance between two atoms. The two atoms are permutable.

1 Angle or scaled angle coordinate between 3 atoms, with the third atom being the vertex. The two side atoms are permutable.

2 Torsion or scaled torsion coordinate.

-1 Tetrahedron out-of-plane coordinate where all four vertices are permutable. The coordinate is defined from the scalar triple product of the three displacement vectors launching from any one vertex **s=r1·(r2xr3)**.

-2 Umbrella out-of-plane coordinate where 3 vertices are permutable, but one other atom is special.
The atom in position 1 is the special atom.
The coordinate is defined from the scalar triple product of the three displacement vector launching from the special atom **s=r1·(r2xr3)**.

Scaling Modes

Scaling Modes for Distance Coordinates

Scaling functions for all supported scaling modes of distance coordinates (*type=0*) are given in the following list (All modes require 2 scaling parameters except for unscaled distance):

0 Unscaled

1 Morse functions **w= exp(-c1∗(r-c2))**

2 Gaussian functions **w=exp(-c1∗(r-c2)∗∗2)**

3 Leading term of screened Columb potential **w=exp(-c1∗(r-c2))/r** (Yukawa potential)

4 Long range term of screened Columb **w=exp(-r/c2)∗(r/c2)∗∗c1**

5 Lennard Jones potentials **w=(c2/r)∗∗c1**

6 Shifted(chasmless) Yukawa **w=exp(-c1∗(r-c2))/(r+c2)**

Scaling Modes for Angle Bends

Supported scaling modes for angle bends (*type=1*) are

0 Unscaled **w=t**. No scaling parameters.

1 Cosine scaling **w=Cos(t)**. No scaling parameters.

2 Distance scaled cosine scaling **w=Cos(t)/(1+exp[c1∗(r1^2+r2^2-c2^2)]**), where **r1** and **r2** are the sides of the angle. Two scaling parameters **c1** and **c2** are required.

Scaling Modes for Torsion Coordinates

Torsion coordinates (*type=2*) are not yet fully implemented.

Scaling Modes for Tetrahedron Out-of-Plane Coordinates

Supported scaling modes for tetrahedron out-of-plane coordinates (*type=-1*) are shown in the following list. 2 scaling parameters are always required. Here the scalar triple product **s=r1·(r2xr3)** and **ri** denotes the array of all *6* internuclear distances.

0 Divide scalar triple product **s** by powers of the product of all *six* distances: **w=s/(Π ri)^C1**

mode>0 Use the product of scaled distances(with the same mode) between each of the 6 atom pairs to scale the scalar triple product
w=s∗Π Scale[ri]

mode<0 Use power of the reciprocal of *sum* of the distances as the scaling factor **w=C1∗s/(Σ rij)^C2**

Scaling Modes for Umbrella OOP Coordinates

Umbrella OOP coordinates (*type=-2*) are similar to tetraheron OOP coordinates. However, they experience different permutational properties due to one of the atoms being treated special permutationally. Like in the case of tetrahedron OOPs, two scaling parameters are always required. The scaling options for umbrella OOPs are listed below. Note that some of these options differ from the tetrahedron case.

0 Reciprocal scaled with 3 distances from the vertex **w=s/(Π ri)^C1**

mode>0 Use product of the *three* scaled distance coordinates as the scaling factor: **w=s∗Σ Scale[rij]**

mode<0 Use the *harmonic mean* of the three scaled distances as scaling factor: **w=s/Π(1/Scale[rij])**
-mode is used as the scaling mode for the distances and the scaling parameters are given to them.

Additional Order Conditions

Other than the maximum total order and total order for each coordinates, a set of *additional conditions* can be used to further restrict the polynomial expansion in a more detailed manner. Every addtional order condition implies one linear inequality constraint on the orders of coordinate sets defined in the previous section.

Each suction condition input occupies one line which contains **nCoordSets+1** integers. The first **nCoordSet** integers specify the multipliers for the order of each of the coordinate sets and the last integer indicates the maximum of the weighed sum of orders.

For example, with 3 coordinate sets, the condition
2 1 0 6
Implies that *(total order of set 1)* ∗2+ *(total order of set 2)* ≤6

Total order of a coordinate set means the sum of the orders of all the coordinates that are defined in a coordinate set, whether it is a coordinate directly specified in the coordinate set definitin section, or a coordinate generated by a symmetry operation.

SEE ALSO

`connect.in`(1) `irrep.in`(1), `potlib`(1), `surfgen`(1), `surfgen.in`(1),

BUGS

Please send bug reports to Xiaolei Zhu ⟨virtualzx@gmail.com⟩