Notes on the usage of the package for solving spectroscopy problems (DISP) and symmetry analysis (SYMM)

Igor Kochikov (igor@kochikov.ru)

Last revision: Dec. 12, 2009

For details of analysis and software, we also recommend to look at the following publications:

1. I. V. Kochikov, G. M. Kuramshina, Moscow University Chemistry Bulletin (Vestnik Moskovskogo Universiteta. Khimiya), 1985, vol.26(4), p.354).
2. A. G. Yagola, I. V. Kochikov, G. M. Kuramshina, Yu. A. Pentin. Inverse problems of vibrational spectroscopy. VSP, Zeist, The Netherlands, 1999.

Contents:

1. The main features of the software package.

The package allows calculation of normal vibrations in the arbitrary system of internal coordinates as well as in Cartesian coordinates. Redundant coordinates are also allowed. Input force constant matrices may be represented in different ways (in Cartesian, internal or symmetry coordinates) and different units (10^6 cm^{-2}, mdyn/Å, Hartree/$Bohr^2$); the package provides automatic transformation of matrices in various coordinate systems and units.

Force constants may be fitted to experiment in the standard way (when matrix elements are directly defined) and using Pulay scaling approach, when the set of scale factors is defined for an a priori given force constant matrix. It is possible to impose constraints fixing certain force constants at the pre-defined values.

It is also possible to simultaneously simulate several isotopic species of a molecule for calculation of the single molecular force field. Different molecules can also be treated simultaneously under condition that certain sets of force constants in both molecules are assumed equivalent. The package also provides experimental frequencies fitting when some of the frequencies are not available from the experiment.

More details are provided in the section related to input (.mol) file structure.

2. Input files format

General configuration data related to the simulation parameters rather than to particular molecule is collected in the separate file DISP.INI. Data describing a molecule are stored in a file with the extension .MOL.

All input files consist of the sections which may follow in any order. Some of the sections are mandatory, while other sections are optional. Each section starts with a header (in brackets) and continues until the next header or until the end of the file. File parsing is based on the keywords which are case insensitive. Blanks and tabs are used as delimiters. Each line of the input text should not exceed 1000 characters.

Empty lines are ignored, as well as contents of each line after the symbols // (similar to C++ programs); this feature allows inserting arbitrary comments into the file. For temporary switching off large piece of input data, it is possible to place the line #if 0 before it and #endif after it; this is also similar to C coding conventions..

2.1. Configuration file DISP.INI

Configuration file DISP.INI contains simulation parameters. It contains three sections; options in each section may follow in any order.

2.1.1. Section [Job]

Type: which problem is solved. May be defined as *Inverse* and *Direct*.
AlphaMax, *AlphaMin*: initial and final values of the regularization parameters. This parameters depend on data normalization; reasonable values may be 1 (maximum) and 10^{-6} (minimum).
AlphaStepFactor: the value by which regularization parameter is multiplied between the optimization steps. Default value is 1/10.
MinGrad: convergence criterion for optimization (gradient norm). A recommended value is something like 10^{-6}.
MaxIter: maximum number of iterations in optimization process. Reasonable value should be in the range of 100 – 10,000.
Method: one of the two possible modes for the force matrix fitting. When Pulay scale factors are used, should be set as *Pulay*; otherwise, *Standard*.
OptiMethod: the choice of optimization technique. At present, the only used technique is conjugate gradient optimization (*Gradient*).

ToTheEnd: in this mode, the choice of regularization parameter is switched off, and the final (smallest) value of this parameter will be used. This mode may be convenient for solving well-posed problems (or to determine incompatibility measure).

2.1.2. Section [Units]

This section defines units in which output results are generated. Input data do not need to be specified in these units.

Length corresponds to the length unit, and may be set as *Angstrom* or *Bohr*. By default, angstrom units are used.
Force specifies measurement units for the force constants. It may have three values:
1) *Cm* – the matrix is given in 10^6 см$^{-2}$, as it was often done in the earlier Russian publications.
2) *MDyn* – matrix is given in mdyn/Å, mdyn, and mdyn·Å (depending on coordinates being linear or angular).
3) *Hartree* – matrix is given in Hartree/Bohr^2, Hartree/Bohr, Hartree (depending on coordinates). Quantum chemical Cartesian matrices are usually provided in these units (Hartree/Bohr^2).

2.1.3. Section [Options]

UseIntFile: this option allows to use the file of intermediate results that is created in the course of optimization and allows to abort and continue optimization process. May be defined as *YES* (the file will be used if present; *NO* (will not be used); or *ASK* (the user will be asked whether to use the file).

All remaining options of this section may be either *TRUE* or *FALSE*.

UseSymmetry: Enables or disables the use of molecular symmetry if data file contains such information.
UseConstraints: Enables or disables constraints on the force constants if such constraints are present in data file.
UseEquivalencies: Enables or disables equivalence constraints on the force constants of different molecules, if such constraints are present in data file (.pck); find more details in thse section for simultaneous treatment of different molecules.
MinimizeOffDiagonal: In redundant coordinate systems, force constant matrix is ambiguous. When original force matrix is given in Cartesian coordinates, its transformation into redundant internal coordinate system may be done in different ways. Setting this option to *FALSE* we obtain the so-called canonic force field (that is, force constant matrix with a rank equal to the number of independent coordinates). Otherwise, the software will generate force matrix with a minimum off-diagonal norm; this may be preferable from model considerations.
SkipDirectInternal: Usually, simulation is completed by printing out the results in symmetry and internal (or Cartesian) coordinates. For very large molecules, the second problem may require large memory and processing time, so it may be switched off by setting this option to *TRUE*.
VerifyCoordinates: In the input data file, it is possible to specify the user-defined values of internal coordinates. When this option is switched on, the software will check Cartesian coordinates of the atoms and (if necessary) will correct them to correspond to the specified values.

2.1.4. Section [Print Options]

This section defines contents of the output file.

PrintInput: Print input data before solving the problem.
PrintCartesian: If simulation is done in internal coordinates, it is also possible to make printout of the force matrix in Cartesian coordinates.
PrintInternal: For the large sparse matrices, it may be convenient to output matrices as a list of force constant values with the set of coordinate pairs corresponding to this value.
PrintCyvin: Add printout of the matrix of average values of internal coordinates <q_iq_j> introduced by Cyvin.
PrintGMatrix: Requires printing of kinematic coefficients (kinetic energy matrix in momentum representation).
PrintGMatrixSymm: The same as previous for each symmetry block.
PrintModes: Pront vibrational modes for each normal vibration.
PrintModeSymms: The same as previous for each symmetry block.
PrintPED: Print distribution of potential energy between the coordinates for each normal mode.
PrintPEDSymm: The same as previous for each symmetry block.
DumpBinary: used for saving the resulting force constant matrix in binary format, for subsequent loading and re-use. This option is used for the huge matrices when text output becomes too bulky.
DumpCartesian: The same as previous, but used for saving force matrix in Cartesian coordinates.

2.1.5. A sample file DISP.INI

```
[Job]
Type=direct
AlphaMax=1.000000e+04
AlphaMin=1.000000e-04
MinGrad=1.000000e-6
MaxIter=100
Method=Standard         // may also be Pulay
ToTheEnd=FALSE

[Units]
Length=Angstroms        // set by default
Force=MDyn

[Options]
UseIntFile=ASK                // YES, NO or ASK
UseSymmetry=TRUE
UseConstraints=TRUE
MinimizeOffDiagonal=TRUE
SkipDirectInternal=FALSE
VerifyCoordinates=TRUE

[Print Options]
PrintInput=TRUE
PrintGMatrix=TRUE
PrintModes=TRUE
PrintGMatrixSymm=TRUE
PrintModesSymm=TRUE
PrintPED=TRUE
PrintCartesian=TRUE
PrintInternal=FALSE
PrintCyvin=TRUE
DumpCartesian=FALSE
DumpBinary=FALSE
```

2.2. Data file format (.mol)*

In this section we describe the standardized file of the input data, independent of the type of computer used. Many sections of this file contain lists which have the same structure. The *list* starts with defining number of its elements (*Count* = ...), which is followed by the enumerated lines.

2.2.1. Section [Isotopes]

This section is used to set molecule name to be used in output (so it is required even if only one molecule is considered). It also allows to specify different isotopic species of a molecule to be processed together (under assumption they share common force field).

Example for one molecule:

```
[Isotopes]
Count=1
1 C60-4 linear chain
```

Example for several isotopic species sharing the same force field:

```
[Isotopes]
Count=3
1 H2O ordinary water
2 D2O heavy water
3 T2O if you can get it
```

2.2.2. Section [Atoms]

This section is used to set number of atoms in a molecule, their Cartesian coordinates, atomic masses and user-defined names to be used in the report.

The section starts with length unit definition (*Length=*) which may be either *Angstrom* or *Bohr* (see also 2.1.2). By default, units are angstroms.

The rest of this section is the list of atoms; each atom is characterized by its (x,y,z) coordinates, chemical symbol and atomic mass.

Example:

```
[Atoms]
Length=Angstrom   // unnecessary line
Count=180
//No       x        y         z       Name  Mass(amu)
  1   0.000000  3.501544  -7.901932   C    12.011
  2   3.501544  0.701156  -8.603088   C    12.011
  3   0.785000  0.000000  -5.101544   C    12.011
... more atoms
```

If masses are given as mass numbers (without decimal point), the software will replace these values by the accurate masses of the isotopes with the specified mass number.

If the file contains data for different isotopic species of a molecule, atomic names and masses should be specified for each instance (coordinates may be omitted):

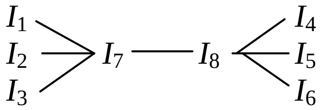
```
[Atoms]
Length=Angstrom
Count=3
//No      x         y          z       Name   Mass(amu)
  1   0.000000  3.501544  -7.901932   H         1.0
                                      D         2.0
                                      T         3.0
  2   3.501544  0.701156  -8.603088   C        12.0
                                      C        12.0
                                      C        12.0
... more atoms
```

2.2.3. Section [Coordinates]

Section Coordinates allows to introduce the system of internal or Cartesian coordinates to be used in calculations. For Cartesian coordinates (*Type* = *Cartesian*), no additional information is required; the software will introduce 3*N* coordinates and assign names for them. For internal coordinates (*Type* = *Internal*), the user provides a list of coordinates data. Each coordinate should be given a name (arbitrary text), type and the list of atoms participating in coordinate definition. In the following text we denote elements of atoms list as I_k.

The following internal coordinate types are considered:

Type 1: coordinates for bond stretching; a pair of atomic indices I_1, I_2 is given which form the bond.
Type 2: coordinates for valence angle bending; the ordered atomic triple I_1, I_2, I_3 is given, the first index corresponding to the atom in the angle vertex.
Type 3: the coordinate for changing the angle between bond I_1I_2 and the plane of atoms $I_2I_3I_4$; for example, the outcome of bond I_1I_2 from the plane $I_2I_3I_4$.
Type 4: the coordinate for changing the dihedral angle formed by atoms $I_1I_2I_3$ and $I_2I_3I_4$, i.e., when the rotational axis passes through the bond I_2-I_3.
Type 5: the coordinate for changing the dihedral angle between two atomic triples $I_1I_2I_3$ and $I_4I_5I_6$. The rotational axis passes through the line of planar intersection in the equilibrium molecular configuration. If all the atoms lie in the same plane, then the rotational axis coincides with the bond I_2-I_5.
Type 6: the coordinate of the torsion of one atomic triple ($I_1I_2I_3$) relative to another ($I_4I_5I_6$) around the bond I_7 - I_8:



If any of the two tops contains less than three atoms, the numbers of missing atoms are denoted as 0. It is required to have at least one atom in each triple.

Example:

```
[Coordinates]
Type=Internal
Count=838
//No Name      Type      Atoms
   1 R1-10       1       1   10             // bond is define by two atoms
   2 R2-19       1       2   19
   3 R3-34       1       3   34
```

```
      ...
 118 A28        2        28   30   11       // angle is defined by three atoms
 119 A29        2        29   31   19
      ... more coordinates
```

Angular coordinate in linear molecules represents a special case. In this case, molecule bending is described by two coordinates instead of one; these two coordinates must follow each other in the coordinate list, and the set of atoms in them should be identical. Two orthogonal deformations will be automatically generated by the software; if it is desirable to directly specify the directions of these deformations, coordinate description is completed by the normal vector coordinates (so that bending takes place in a plane defined by molecular axis and a given vector). Example:

```
//No Name       Type      Atoms
 3 A123x         2        2   1   3   1.0 0.0 0.0
 4 A123y         2        2   1   3   0.0 1.0 0.0
```

Here we assume that molecular axis is aligned with Cartesian *z*-axis.

Finally, this section may be used for adjusting atomic coordinates. Suppose that you have only approximate Cartesian coordinates of atoms, but the values of internal coordinates for accurate configuration are known. In this case, it is possible to specify desired value of internal coordinate after description of atomic pairs or triples (lengths are specified in angstroms, and angles in degrees). Example:

```
  1 R1-10      1        1   10            Value=1.490
  2 R2-19      1        2   19            Value=1.490
  3 R3-34      1        3   34            Value=1.490
      ...
 118 A28        2        28   30   11       Value=120.0
 119 A29        2        29   31   19       Value=120.0
```

If the file DISP.INI contains option *VerifyCoordinates = TRUE*, Cartesian coordinates of all atoms will be adjusted so as to correspond to the desired values of internal coordinates. If the values are incompatible, the least-squares solution will be generated.

An additional option *Type = Auto* will force the software to generate a system of internal coordinates consisting of bond lengths and valence angles. Bonds will be generated depending on the maximum bond length specified by the user, and valence angles will be created for all pairs of the adjacent bonds. Since the constructed set of coordinates is often insufficient for complete description of molecular geometry, this option is recommended for use with SYMM package that will insert the constructed coordinates into the original MOL file; later on it can be manually extended to provide missing coordinates. Example:

```
[Coordinates]
Type=Auto
Count=2
//No Pair   Max Distance
  1  C-C      2.0
  2  C-H      1.4
```

2.2.4. Section [Symmetry]

This section is optional. If it is missing, the software issues a warning and may continue simulation assuming C_1 symmetry. Even when this section is present, symmetry properties may be ignored by setting *UseSymmetry* = *FALSE* in the configuration file DISP.INI.

The software package SYMM can automatically amnalyze molecular symmetry and insert appropriate [Symmetry] section into the original .MOL file, so it's rarely has to be created manually.

The main contents of this section is a matrix for transfer from internal (or Cartesian) to symmetry coordinates. The matrix should have the same dimensions as the number of the introduced coordinates. The following variants are implemented:

Matrix =*Unit* means that molecule possesses no symmetry (or when all coordinates are at the same time symmetry coordinates).
Matrix =*Text* means that matrix elements are to be found in the following lines of the input file. If there are complex coefficients, this line should also contain the word *Complex* after *Text* (delimited by comma or space).
Matrix =*File* means that matrix is to be read from a binary file (see example below). This way is convenient for huge matrices generated by the symmetry analysis software.

Matrix in text format does not need to be normalized (it will be normalized automatically), and may contain any number of elements in each line; if there are too many elements in a row, the matrix row may be continued in the next line of input data. To represent complex values that may appear in the symmetry matrix, special symbols are used, for example:
c and *d* (the two complex cubic roots of 1);
i (square root of -1).
Complete list of symbols is shown in SYMM software report).

After the matrix, the section contains the list of all symmetry types of molecular vibrations with the number of independent degrees of freedom (frequencies) and total number of symmetry coordinates of this type (which may contain dependent coordinates).

Symmetry block names (except the first character) may be arbitrary. The first character defines how many identical blocks of this type should be generated; for the blocks with multiplicity 2, 3, 4, 5 the letters E, F, G and H are reserved.

The order of symmetry types must coincide with the order of symmetry coordinates input in the matrix of reduction by the symmetry.

Example of a matrix in text format:

```
[Symmetry]
Matrix=Text
    1    0    0    0    0    0    0    0    0    0    0    0    0
    0    1    1    1    1    0    0    0    0    0    0    0    0
    0    0    0    0    0    1    1    1    1    0    0    0    0
    0    0    0    0    0    0    0    0    0    1    1    1    1
    0    1   -1    1   -1    0    0    0    0    0    0    0    0
    0    0    0    0    0    1   -1    1   -1    0    0    0    0
    0    0    0    0    0    0    0    0    0    1   -1    1   -1
    0    1    0   -1    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    1    0   -1    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    1   -1   -1    1
    0    0    1    0   -1    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    1    0   -1    0    0    0    0
```

```
   0   0   0   0   0   0   0   0   0   1   1  -1  -1
BlocksCount=4
//Name  Frequencies Coordinates
1   A1         3           4
2   B1         2           2
3   B2         1           1
4   E          3           3
```

Example where symmetry matrix is read from a binary file:

```
[Symmetry]
Matrix=File:c60-3.cmx
BlocksCount=8
//Name  Frequencies Coordinates
1   Ag        71         118
2   Au        64          94
3   B1g       64          96
4   B1u       69         114
5   B2g       66         101
6   B2u       67         107
7   B3g       66         104
8   B3u       67         104
```

In this case, matrix dimensions are 838 by 838, and its text representation would be quite inconvenient.

2.2.5. Section [Frequencies]

This section contains experimentally measured frequencies of normal vibrations and is necessary only for solving inverse problems. It is essentially a list of frequencies (with symmetry type specified) and estimates of experimental error.
Both frequencies and errors are measured in cm^{-1}.

Example:

```
[Frequencies]
Count=20
//  Symm.  Freq.   Error
1   A1g    3074.00   3.00
2   A1g     993.00   3.00
3   A2g    1350.00   3.00
4   A2u     674.00   3.00
... more frequencies
```

For a set of isotopic species, frequencies are specified in the following columns of the same list:

```
[Frequencies]
Count=20
//        Simple benzene   Deuterated benzene
//  Symm.  Freq.   Error       Freq.   Error
1  A1g  3074.00   3.00    2303.00   3.00
2  A1g   993.00   3.00     946.00   3.00
3  A2g  1350.00   3.00    1059.00   3.00
4  A2u   674.00   3.00     496.00   3.00
5  B1u  3057.00   3.00    2285.00   3.00
... more frequencies
```

In general, [Frequencies] section may be accompanied by other sections describing other kinds of experimental data to be processed; here we don’t describe them.

2.2.6. Section [Matrix Z] ([Matrix F])

This section is used to input initial approximation of the force constant matrix. Representation of this matrix (and measurement units) may differ from that set in configuration file (for example, one may solve the problem in internal coordinates and have an input matrix in Cartesian coordinates, which is often the case when using ab initio results).

Force matrix units are specified in the line (*Units=...*) where possible values are the same as those described in 2.1.2.

The force matrix itself may be represented in one of the following ways:

1) *Coordinates=Cartesian*
The matrix is present in triangular form. In fact, the lower triangle of a matrix is split into columns of arbitrary size, and these columns are entered one after another. An example of such input mode follows:

```
[Matrix Z]
Coordinates=Cartesian
Units=Hartree

    1-x   0.7744
    1-y  -0.0056  0.7357
    1-z  -0.0000  0.0000  0.1530
    2-x  -0.2814 -0.1256 -0.0000  0.8052
    2-y  -0.0382 -0.2088 -0.0000 -0.0537  0.7674
    2-z  -0.0000 -0.0000 -0.0747  0.0000  0.0000  0.1708
    3-x   0.0744 -0.0138 -0.0000 -0.1688 -0.0345 -0.0000  0.7976
    3-y  -0.0945 -0.0434  0.0000  0.0703 -0.3823 -0.0000  0.0598  0.8022
    3-z  -0.0000  0.0000  0.0071  0.0000 -0.0000 -0.0708 -0.0000 -0.0000
    ... (several lines omitted)...
   14-z  -0.0000 -0.0000  0.0210  0.0000  0.0000  0.0024 -0.0000 -0.0000
          1-x     1-y     1-z     2-x     2-y     2-z     3-x     3-y
    3-z   0.1502
    4-x  -0.0000  0.7707
    4-y   0.0000  0.0058  0.7853
    4-z  -0.0799  0.0000  0.0000  0.1904
    5-x   0.0000 -0.3172 -0.1464 -0.0000  0.8331
    5-y   0.0000 -0.0201 -0.1988  0.0000 -0.0481  0.7672
    5-z   0.0089 -0.0000 -0.0000 -0.0797 -0.0000 -0.0000  0.1476
    6-x   0.0000  0.0860 -0.0214 -0.0000 -0.1671 -0.0242  0.0000  0.7907
    ... (several lines omitted)...
   14-z  -0.0010  0.0000  0.0000  0.0003  0.0000 -0.0000 -0.0005  0.0000
           3-z     4-x     4-y     4-z     5-x     5-y     5-z     6-x
    ... (many lines omitted)...
   14-y   0.1494
   14-z  -0.0000  0.0273
           14-y    14-z
```

2) *Coordinates=Internal*
is completely analogous to the previous case. It is assumed that rows and columns of a matrix correspond to the list of coordinates introduced in the [Coordinates] section.

Both Cartesian and internal matrices may be accepted from an external binary file. These files may be generated on demand by DISP program (see 2.1.3). Example of file input:

```
[Matrix Z]
Coordinates=Cartesian
Units=Hartree
```

```
File=car-02.dmp
```

3) Coordinates=Symmetry
is similar to the previous, but all matrices for each symmetry block are listed one after another. The number of blocks and their size and order should corresponmd to the description in [Symmetry] section. Example:

```
[Matrix Z]
Coordinates=Symmetry
Units=Cm
Block=A1
 8.4620
 -0.0497 11.9070
-0.1476 -0.6315 5.3850
 0.0000  0.0000 0.0000 0.0000
Block=E
 7.3750
 0.3624  1.1702
 0.0921 -0.4248 2.3871
```

4) Input as a list
All matrices in case they are sparse (contain many zeroes) which may happen while modeling, may be represented by the list of non-zero elements. In this case, *Coordinates=* line should include the keyword *List*, and each line of the list should refer to the pair of coordinate indices corresponding to the particulare force constant. Example:

```
[Matrix Z]
Coordinates=Internal(List)
Units=Cm
  7.0000    1,1
  0.2039    1,2   1,3   1,4   1,11  1,12  1,13
  0.9604    11,13 12,13
 12.0000    2,2   3,3   4,4   11,11 12,12 13,13
  0.6178    1,14  1,15  1,16
  0.9604    2,3   2,4   3,4   11,12
  0.6178    1,5   1,6   1,7
  0.4632    2,5   3,6   4,7   11,14 12,15 13,16
```

When symmetry is present, it is not necessary to specify all matrix elements; the nonzero values will be assigned to all elements equivalent by symmetry. Such a list may be generated by DISP program on demand; see (2.1.3) for more information.

2.2.7. Section [Matrix Z0] ([Matrix F0])

This matrix is used as a stabilizer when solving the inverse problem. Its input is absolutely the same as described in the previous section. In most cases it is convenient to use identical matrices of stabilizer and of initial approximation; therefore when one of the sections [Matrix Z], [Matrix Z0] is missing, it is assumed that these two matrices are identical.

2.2.8. Section [Constraints]

This section is used when solution of the inverse (or direct) problem is subject to constraints. Here we only described the equivalence constraints that force some matrix elements to be equal to each other or to zero. The set of constrained elements may be specified explicitly ot implicitly. It is defined by the type of constraints (*Action=*) and set of elements (*Set=*). It is allowed to introduce several different constraint with their own sets of elements.

Action=Fix corresponds to fixing the force constants at their initial values.
Action=Fix-At-0 means that certain force constants will be kept at zero values.
Action=Release means that previously introduced constraints are discarded. It is sometimes convenient to introduce constraints on a wide set of the elements, and then remove them for the smaller set.

The set of elements (*Set=*) may be defined as follows:

All – apply constraints on the all matrix elements.
Diag – apply constraints to the diagonal matrix elements. This type of constraint is more often used with (*Action=Release*).
OffDiag – apply constraints to off-diagonal elements.
Distant – apply constraints to the elements of matrix that correspond to distant molecular fragments. Matrix element is participating in the constraints set if a pair of internal coordinates corresponding to this matrix element does not contain common atoms in definition of the coordinates (in particular, the elements of matrix *G* in this case are equal to zero).
Far – the same as *Distant*, but a pair of coordinates should not have either common, or adjacent atoms. This is a subset of *Distant*.
Zeroes – constraints will be applied to matrix elements that are equal to zero in the initial matrix (and hence will remain zero during optimization).
List – this means that the following lines of the input file contain explicit description of the matrix elements; their description is a list of coordinate pairs delimited by commas.

Example of constraints:

```
[Constraints]
Action=Fix        Set=OffDiag
Action=Fix-at-0   Set=List
10,1  10,2  10,3  10,4  10,5  10,6  10,7  11,1  11,2
11,3  11,4  11,5  11,6  11,7
Action=Release    Set=List
5,4   6,4
```

In this example, we first fix all off-diagonal matrix elements at their initial values, then fix some of the elements (given by the list) at zero values, and then include into optimization procedure some previously fixed elements.

2.2.9. Section [Amplitudes]

This is a purely informative section used for calculation and output of some values that are not usually present in the report – namely, mean square amplitudes of interatomic distances. The section contains the list of atomic pairs for which the amplitudes should be defined, and temperatures at which the amplitudes should be calculated. Example:

```
[Amplitudes]
Count=3
  1-3         2-3         1-2
Temperatures
0.000000  300.000000  400.000000
```

2.2.10. Section [Pulay Factors]

In the mode when Pulay factors are to be defined rather than force matrix elements themselves, this section allows to set initial values of scale factors and introduce certain constraints which will be supported during optimization. If this section is absent, initial values of all scale factors will be set to 1, and all of them will be allowed to vary independently.

The usage of this section can be understood from the following example:

```
[Pulay Factors]
Count=17
// attribute = Fix for fixed factor, equal for equal factors
// may be absent if it is the only one
//No Name Value Attribute
   1 Q    1.10      Fix
   2 Q1   1.00      Q
   3 Q2   1.00      Q
   4 q1   1.00      q
   5 q2   1.00      q
   6 q3   1.00      q
   7 G1   1.00      Fix
   8 G2   0.90      Fix
   9 G3   1.00      Fix
  10 B1   1.00      B
  11 B2   1.00      B
  12 B3   1.00      B
  13 A12  1.00      A
  14 A13  1.00      A
  15 A23  1.00      A
  16 X    1.00
  17 T    1.00
```

Pulay factors with the same *Attribute* will be kept equal during the optimization. When *Attribute* = *Fix*, this factor will not be changed at all (remaining at its initial value). Coordinate names in the list should correspond to that introduced in [Coordinates] section. Attribute names are arbitrary (except the reserved word *Fix*).

2.2.11. Section [User Coordinates].

This section is useful if it is desired to run simulation in some special coordinate system - arbitrary user coordinates. The software allows to introduce any coordinates that are represented as linear combinations of the internal coordinates. On this way, for example, it is possible to introduce the coordinates of local symmetry.

The syntax of the section is quite simple: linear combinations are written explicitly. The coefficients may be arbitrary integer of real numbers. The names of user coordinates (1st column) may be arbitrary (in particular, they may coincide with the names of internal coordinates). All coordinate names appering in the right-hand side, should be defined in [Coordinates] section of the input file.

Example of a recommended set of local symmetry coordinates for a simple XYZ_3 (C_{3v}) molecule:

```
[User Coordinates]
Count=9
  1 Q   =   Q
  2 q1  =   q1
  3 q2  =   q2
  4 q3  =   q3
  5 sd  =   a1+a2+a3-b1-b2-b3
  6 ad1 =   2a1-a2-a3
  7 ad2 =   a2-a3
  8 r1  =   2b1-b2-b3
  9 r2  =   b2-b3
```

Note that number of user coordinates may be smaller than the number of internal coordinates. Input matrices in .MOL file may be specified in any coordinate system (including User coordinates if they are introduced). Output files will be given in both user and internal coordinates.

Please bear in mind that user coordinates may have different symmetry compared to internal coordinates. Therefore, introduction of user coordinates requires re-running SYMM package for obtaining proper information on the symmetry coordinates.

2.2.12. Sample input file.

For convenience, we provide an example of the file constructed for calculating the force field of a relatively small molecule, namely, WOF_4.

This molecule belongs to the point symmetry group C_{4v} and has the following geometric parameters in equilibrium configuration: *r* (W-O) = 1.666 Å, *r* (W-F)=1.847 Å, angle O-F-W = 104.8 degrees. As for the matrix F^0 determining the Tikhonov functional stabilizer, we have used the model diagonal matrix. The experimental frequencies used for calculations were taken from the gas phase spectra; the error of their measurement was assumed to be ~ 1 cm^{-1}.

```
 [Isotopes]
Count=1
// List of molecule names
1  W-O-F4

[Atoms]
Length=Angstroms
Count=6
//No     x          y          z        Name   Mass(a.m.u.)
  1   1.785723   0.000000  -0.471808   1-F       18.998
  2   0.000000   1.785623  -0.471808   2-F       18.998
  3  -1.785723   0.000000  -0.471808   3-F       18.998
  4   0.000000  -1.785623  -0.471808   4-F       18.998
  5   0.000000   0.000000   1.666000   5-O       16.000
  6   0.000000   0.000000   0.000000   6-W      183.850

[Coordinates]
Type=Internal
Count=13
//No Name    Type     Atoms
   1 Q          1       5    6
   2 q1         1       1    6
   3 q2         1       2    6
   4 q3         1       3    6
   5 q4         1       4    6
   6 b1         2       6    1    5
   7 b2         2       6    2    5
   8 b3         2       6    3    5
   9 b4         2       6    4    5
  10 a12        2       6    1    2
  11 a23        2       6    2    3
  12 a34        2       6    3    4
  13 a41        2       6    4    1
```

```
[Symmetry]
Matrix=Text
  1  0  0  0  0  0  0  0  0  0  0  0  0
  0  1  1  1  1  0  0  0  0  0  0  0  0
  0  0  0  0  0  1  1  1  1  0  0  0  0
  0  0  0  0  0  0  0  0  0  1  1  1  1
  0  1 -1  1 -1  0  0  0  0  0  0  0  0
  0  0  0  0  0  1 -1  1 -1  0  0  0  0
  0  0  0  0  0  0  0  0  0  1 -1  1 -1
  0  1  0 -1  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  1  0 -1  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  1 -1 -1  1
  0  0  1  0 -1  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  1  0 -1  0  0  0  0
  0  0  0  0  0  0  0  0  0  1  1 -1 -1
BlocksCount=4
//Name  Frequencies Coordinates
1   A1         3           4
2   B1         2           2
3   B2         1           1
4   E          3           3

[Frequencies]
Count=9
//  Symm.  Freq.   Error
  1  A1   1056.00   1.00
  2  A1    733.00   1.00
  3  A1    248.00   1.00
  4  B1    631.00   1.00
  5  B1    291.00   1.00
  6  B2    328.00   1.00
  7  E     698.00   1.00
  8  E     300.00   1.00
  9  E     235.00   1.00

[Matrix Z]
Coordinates=Symmetry
Units=Cm
Block=A1
     S1  16.2900
     S2   0.0000  8.4500
     S3   0.0000  0.0000  1.5910
     S4   0.0000  0.0000  0.0000  1.6870
           S1      S2      S3      S4
Block=B1
     S5   8.1450
     S6   0.0000  1.5910
           S5      S6
Block=B2
     S7   1.6870
           S7
Block=E
     S8   8.1450
     S9   0.0000  1.5910
    S10   0.0000  0.0000  1.6870
           S8      S9      S10

[Matrix Z0]
Coordinates=Symmetry
Units=Cm
Block=A1
     S1  16.2900
     S2   0.0000  8.4500
     S3   0.0000  0.0000  1.5910
     S4   0.0000  0.0000  0.0000  1.6870
           S1      S2      S3      S4
Block=B1
     S5   8.1450
     S6   0.0000  1.5910
           S5      S6
Block=B2
     S7   1.6870
           S7
Block=E
     S8   8.1450
     S9   0.0000  1.5910
    S10   0.0000  0.0000  1.6870
           S8      S9      S10

[Amplitudes]
Count=5
  6-5     6-1     5-1     1-2     1-3
Temperatures
0.000000  300.000000  400.000000
```

2.3. Common treatment of several molecules.

For simultaneous processing of two or more molecules, a new input file (.PCK) is created; this file contains data on the molecules to be processed and describes the links between the molecular force fields. Contents of this file is described in this section.

2.3.1. Section [Molecules]

This section must be present. It contains the list of .MOL files, each of them describing one of the participating molecules. Example:

```
[Molecules]
Count=4
1. adam-td.mol
2. adam-c3v.mol
3. adam-c2v.mol
4. adam-cs.mol
```

2.3.2. Section [Links]

Simultaneous treatment of many molecules has sense only when we introduce some constraints that connect matrix elements of different molecules; otherwise the results will be the same as for separate sequential treatment of these molecules.

Links between molecular force fields are introduced similar to constraints. In this case, we also specify action (*Action*=) which may be either *Fix*, or *Release*, and describe a set of elements to which we want to apply this kind of constraint (*Set*=).

Here are the allowed values of *Set*=:

All – all force matrix elements are kept equal. Naturally, this kind of constrains may be applied only when force matrices have the same size (and, generally, when coordinates have the same meaning). It may have sense to use this type of constraint when processing isotopic species of a molecule with the same geometry but different symmetry.

List – here we explicitly describe all matrix elements that should be kept equal. In the following line, for example, we introduce equivalence between element (2,2) of a molecule 1 and element (3,3) of the molecules 2 and 3::

```
Action=Fix Set=List
1(2,2) = 2(3,3) = 3(3,3)
```

Another example shows how to set equivalence for multiple elements.

```
Action=Fix Set=All
Action=Release Set=List
1(1,*)
2(2,*)
```

Here we start with introducing equivalencies for all matrix elements; the second line excludes from the list of equivalencies all elements of the first row and columns for molecule 1 and elements of the second row and column for the molecule 2.