æ

MOLVIB

Version 7.0

User's Guide

Tom Sundius

Helsinki, July 2000

1. Introduction

Revisions made : 1978-11-02 , 1979-04-23, 1979-11-09, 1983-01-18, 1984-08-06, 1988-11-21, 1991-01-14, 1997-06-13, 1999-10-23, corr. 2019-5-4.

MOLVIB is a Fortran program for the calculation of harmonic force fields and vibrational modes of molecules with up to 30 atoms (standard version).

Normal coordinate analysis is nowadays commonly employed as an aid in the interpretation of the vibrational spectra of large molecules. In order to get meaningful results, a knowledge of the vibrational force field is necessary. Since the number of force constants grows quadratically with the number of atoms, one has to employ many approximations in the calculation of harmonic force fields even for moderately large molecules.

To overcome this difficulty one can determine a force field for a set of related molecules using the so-called overlay method introduced by Snyder and Schachtschneider in the 1960's [17].

About 1970 Gwinn developed a program for normal coordinate analysis using mass-weighted cartesian coordinates [8], which eliminates the redundancy problems arising when internal valence coordinates are used, as in Wilson's GF-method. MOLVIB is based on the same fundamental idea, but differs from similar programs in many respects. The program was first described by Sundius [20], and a more recent description by the same author can be found in ref [21].

In addition to free molecules, crystals can also be treated. In this case, up to 50 atoms divided between 11 sub-units can be handled. All the calculations are performed in terms of mass-weighted cartesian coordinates, instead of internal coordinates as in the conventional GF-method. This makes it possible to overcome problems with redundant coordinates. The force field is refined by a modified least squares fit of the observed normal frequencies, as described by Sundius [22]. Beginning with version 6 of the program, it is possible to change several of the maximum array dimensions, and create executables, which can handle larger molecules (or force fields).

The program is user-friendly, and can be easily adopted to different force fields. It is also possible to express the force field in the CFF notation (see the book by S.R. Niketić and Kj. Rasmussen [13]). This makes it possible to use it in combination with the CFF program (or other molecular mechanics programs) for conformational analysis of flexible molecules.

2. Theoretical background

In the following, we will give a short description of theoretical foundations of the program.

2.1 Classical theory of molecular vibrations

The potential energy of a molecule with N atoms can be expressed as an analytical function in 3N cartesian atomic coordinates $R_1, R_2, \ldots, R_{3N}$:

$V(R) = V(R_1, R_2, \ldots, R_{3N})$.

Let us now assume that the atomic nuclei perform *small* oscillations about their equilibrium positions (large amplitude motion is not considered here), defined by the vector

$R^0 = (R_1^0, R_2^0, \ldots, R_{3N}^0)$. Then the potential energy $V(R)$ may be expanded in a power series in terms of the cartesian *displacement coordinates* $x_i = R_i - R_i^0$ $(i = 1, 2, \ldots, 3N)$:

$$V(R) = V(R^0) + \sum_i \left(\frac{\partial V}{\partial x_i}\right)_0 x_i + \frac{1}{2}\sum_{i,j}\left(\frac{\partial^2 V}{\partial x_i \partial x_j}\right)_0 x_i x_j + \ldots$$

We now consider the equilibrium energy $V(R^0)$ as a zero point of energy and use the condition of equilibrium

$$\left(\frac{\partial V}{\partial x_i}\right)_0 = 0, \qquad i = 1, 2, \ldots, 3N.$$

In the *harmonic* approximation, when all terms of the third order or higher are neglected, the potential energy may thus be expressed as

$$V = \frac{1}{2}\sum_{i,j} f_{ij} x_i x_j,$$

where the coefficients f_{ij}, the second order partial derivatives of the potential energy in equilibrium, form a square-symmetric matrix of order $3N$ (called the *force constant* matrix F).

In the classical description, the total kinetic energy of the N atomic nuclei may be written

$$T = \frac{1}{2}\sum_i M_i \dot{x}_i^2.$$

Substituting the expressions for the potential and kinetic energies into Lagrange's equations

$$\frac{\partial V}{\partial x_i} + \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{x}_i}\right) = 0 \qquad (i = 1, 2, \ldots, 3N),$$

we get the classical Newton equations of motion:

$$M_i \ddot{x}_i + \sum_j f_{ij} x_j = 0 \qquad (i = 1, 2, \ldots, 3N).$$

These equations have the periodic solutions

$$x_i = x_i^0 \sin(\omega t + \delta) \qquad (i = 1, 2, \ldots, 3N).$$

Substituting the solutions back into the equations of motion we find a new set of equations, from which the amplitudes x_i^0 and the *normal frequencies* ω may be determined:

$$\sum_j f_{ij} x_j^0 = M_i \omega^2 x_i^0 \qquad (i = 1, 2, \ldots, 3N).$$

This set of equations is equivalent with the eigenvalue equation for the force constant matrix F.

Often it is more convenient to use mass weighted cartesian displacement coordinates than ordinary displacement coordinates. The mass weighted coordinates are defined by the relations

$$\xi_i = \sqrt{M_i} x_i \qquad (i = 1, 2, \ldots, 3N).$$

The above equations can then be written in the form

$$\sum_j (M_i M_j)^{-1/2} f_{ij} \xi_j^0 = \omega^2 \xi_i^0 \qquad (i = 1, 2, \ldots, 3N),$$

which is equivalent with the eigenvalue equation for the *dynamical matrix*

$$D = M^{-1/2} F M^{-1/2},$$

where $M^{-1/2}$ is a diagonal matrix containing the reciprocal square roots of the atomic masses.

This matrix, which is of fundamental importance in the theory of lattice vibrations, is very convenient to use in the normal coordinate analysis of isotopically substituted molecules, because it includes the atomic masses in an explicit manner. The eigenvalue equation for the dynamical matrix can be expressed in matrix form as

$$\tilde{U} D U = \Lambda,$$

where Λ is a diagonal matrix consisting of the eigenvalues $\lambda_i = \omega_i^2 \quad (i = 1, 2, \ldots, 3N)$ and U is an orthogonal matrix containing the normalized eigenvectors.

Because a molecule in three-dimensional space has $3N - 6$ degrees of freedom, six of the eigenvalues of the dynamical matrix will usually be zero (a proof can be found, e.g., in Califano [5], Ch. 2.6). This is a very useful test of the accuracy of the calculations.

If the atomic masses are expressed in atomic mass units, the frequencies in cm^{-1}, and the energy in the unit aJ ($= 10^{-18}$J), then it can be shown that the relation between the *normal frequency* of vibration and the corresponding eigenvalue of the dynamical matrix is

$$\nu_i = 1302.79 \sqrt{\lambda_i}.$$

The orthogonal transformation U defines a new set of coordinates, called *normal coordinates* (or normal modes) $Q_i \quad (i = 1, 2, \ldots, 3N)$:

$$\xi = UQ, \qquad Q = \tilde{U}\xi.$$

In terms of these coordinates the potential and kinetic energies are *diagonal* :

$$V = \frac{1}{2}\tilde{x}Fx = \frac{1}{2}\tilde{\xi}D\xi = \frac{1}{2}\tilde{Q}\tilde{U}DUQ = \frac{1}{2}\tilde{Q}\Lambda Q$$
$$T = \frac{1}{2}\tilde{\dot{x}}M\dot{x} = \frac{1}{2}\tilde{\dot{\xi}}\dot{\xi} = \frac{1}{2}\tilde{\dot{Q}}\tilde{U}U\dot{Q} = \frac{1}{2}\tilde{\dot{Q}}\dot{Q}.$$

Frequently the force constants f_{ij} are not defined in terms of cartesian coordinates, but are rather expressed in *internal coordinates* [23] (or their linear combinations), which are more natural from a chemical point of view, because they can be directly related to molecular bond lengths and bond angles.

The relation between the internal coordinates s_i and the cartesian displacement coordinates x_i can be expressed in matrix form as

$$s = Bx.$$

Usually the matrix B is rectangular, because the number of internal coordinates is seldom identical to the number of cartesian displacement coordinates. This means that it is impossible to invert the transformation matrix. A method to overcome this problem will be presented in sect. 2.8.

If the matrix F is expressed in terms of internal coordinates, the potential energy may be expressed

$$V = \frac{1}{2}\tilde{x}\tilde{B}F_sBx = \frac{1}{2}\tilde{s}F_ss,$$

and the dynamical matrix thus assumes the form

$$D = M^{-1/2}\tilde{B}F_sBM^{-1/2}.$$

The normal modes can be conveniently characterized with the L–matrix, which gives the transformation from normal to internal coordinates. With the aid of the B-matrix and the eigenvector matrix U the L-matrix can be calculated according to the formula

$$L = BM^{-1/2}U.$$

To facilitate the interpretation of the normal modes, the columns are sometimes normalized to 1.

The normal modes can also be characterized by means of the *potential energy distribution* (P.E.D.), which is calculated from the L–matrix and the F–matrix in internal coordinates according to the formula

$$P_{ij} = (L_{ji})^2F_{jj}/\lambda_i.$$

The P.E.D., which gives the fractional contribution of the diagonal matrix elements of F to the normal modes, is used for a complete symmetry classification of the normal vibrations of the molecule.

In MOLVIB these matrices are constructed for each molecule (and a set of symmetry coordinates) in turn. When isotopically substituted molecules are considered, the symmetry coordinates are input only for the normal molecule, since the symmetry coordinates and the F-matrices are the same for the isotopic analogues (cf. sect. 3.3.3).

2.2 Construction of the force constant matrix F.

We shall now describe, how the force constant matrix is constructed for some commonly used model potentials. Let us first introduce a new notation for the F-matrix, which is used in MOLVIB.

Any force constant matrix may be expressed in the form

$$F = Cf = \sum_k C^{(k)} f_k,$$

where $f(\tilde{f} = (f_1, f_2, \ldots, f_m))$ denotes a set of force constants (*force constant vector*) and $C = (C^{(1)}, C^{(2)}, \ldots, C^{(m)})$ is a row "vector" of coefficient matrices. This method of expressing the F-matrix is very convenient in practical force constant calculations, as we shall see later on (see also the book by Califano [5], sect. 8.6).

The C-matrices describe the dependence of the force field on the internal (or symmetry) coordinates. This notation has the advantage that it is very easy to apply linear constraints on the force constants. Assume, e.g., that the matrix F is diagonal with respect to three force constants f_1, f_2 and f_3, and that the relation

$$f_3 = \alpha f_1 + \beta f_2$$

holds. In coefficient matrix form, the F-matrix may then be expressed

$$\begin{pmatrix} f_1 & 0 & 0 \\ 0 & f_2 & 0 \\ 0 & 0 & f_3 \end{pmatrix} = \begin{pmatrix} f_1 & 0 & 0 \\ 0 & f_2 & 0 \\ 0 & 0 & \alpha f_1 + \beta f_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \alpha \end{pmatrix} f_1 + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \beta \end{pmatrix} f_2.$$

Depending on the type of model potential used, it may be more convenient to express the force constant matrix in cartesian than in internal coordinates. As mentioned previously, the translations and rotations of the molecule as a whole will give rise to six normal frequencies, which have the magnitude zero. This means that the redundant coordinates need not to be eliminated before diagonalization, if the eigenvalue problem is solved in cartesian coordinates. This is one of the advantages with force field calculations in cartesian coordinates.

Assume now (as before) that the internal coordinate vector s is linearly related to the cartesian vector x through the matrix relation

$$s = Bx$$

(i.e., we neglect all higher order terms). We then find that

$$F = \tilde{B} F_s B = \tilde{B} C_s f B = \sum_k \tilde{B} C_s^{(k)} B f_k,$$

which means that we can write

$$F = Cf = \sum_k C^{(k)} f_k,$$

where the transformed matrices $C^{(k)}$ are given by

$$C^{(k)} = \tilde{B} C_s^{(k)} B$$

or explicitly

$$C_{ij}^{(k)} = \sum_{l,m} B_{li} B_{mj} (C_s^{(k)})_{lm}.$$

As we have mentioned earlier, the normal vibrations and normal modes are found by diagonalization of the dynamical matrix D, which in the coefficient matrix notation may be written

$$D = M^{-1/2} F M^{-1/2} = M^{-1/2} C f M^{-1/2}.$$

The eigenvalue equation may then be resolved in the following set of linear equations:

$$\sum_{k=1}^{m} (\tilde{U} M^{1/2} C^{(k)} M^{-1/2} U)_{ii} f_k = \lambda_i \ (i = 1, 2, \ldots, 3N).$$

In order to calculate the F-matrix elements, there are several potential models to choose between. The most commonly used models are the *simple valence force field* (SVFF), the (simplified) *general valence force field* (GVFF), the Urey-Bradley force field (UBFF) and the *local symmetry force field* (LSFF) (see sect. 2.4).

Obviously, the most general harmonic potential function in terms of internal coordinates can be expressed as

$$V = \frac{1}{2} \sum_{i,j} f_{ij} s_i s_j.$$

This is a GVFF which includes interactions between *all* the internal coordinates.

For polyatomic molecules, this potential is, however, impractical or even impossible to use as such, because the number of force constants increases quadratically with respect to the number of internal coordinates, which is of the same order of magnitude as the number of normal frequencies. Even if we use the frequencies of deuterated or isotopically substituted molecules in the force constant calculations, the determination of a complete GVFF remains an unattainable goal for most larger molecules.

The simplest possible valence force field is obtained by retaining only the diagonal terms of the GVFF:

$$V = \frac{1}{2} \sum_{i} f_i s_i^2.$$

This is called the *simple valence force field*, which was introduced already in the first normal coordinate calculations by Bjerrum in 1914 [4].

In this case the number of force constants is equal to the number of internal coordinates, which often makes it possible to determine the force constants from the observed frequencies. However, it is very crude, because it does not contain any interactions.

The SVFF may be improved in many ways. One method is to include interactions between neighbouring internal coordinates, such as stretch-stretch, stretch-bend and bend-bend interactions. We then get what is sometimes called a *simplified* GVFF. Many molecules have been treated successfully by this method during the last 20 years. As an example, we could mention the extensive force field calculations on the alkanes performed by Schachtschneider and Snyder [17]. The local symmetry force field, which was introduced by Shimanouchi [18], will be described later in connection with coordinates.

The question is, however, how to choose the interactions that are to be included. In addition, it may be difficult to give the force constants a physical interpretation, since the interactions usually involve several atoms. In this respect the *Urey-Bradley force field* means an improvement. It was introduced by Urey and Bradley in 1931, and later on improved by Shimanouchi and his coworkers [18].

In its usual formulation, the UBFF includes interactions between atoms bonded to a common atom (the so-called 1-3 interactions) in addition to the diagonal valence force field terms. This means that the interactions are *localized* and interpretable in terms of an atom-atom interaction potential. In the UBFF, the forces between non-bonded atoms usually are assumed to be repulsive and inversely proportional to a power of the distance between the atoms. The F-constants of the Urey-Bradley force field, which are associated with the interactions between every two atoms bonded to a common atom, are defined by the bending coordinate of the angle between two bonds. The elimination of the redundant Δq-coordinates is done in MOLVIB by the subroutine UREY, which is presently restricted to treat only geminal (1,3) interactions. If NBTYP=0, the F'-constants of the Urey-Bradley field can be varied independently from the F-constants. With NBTYP=1, the ratio F'/F is assumed to be fixed.

In molecular mechanics calculations, the force field (potential energy) is parametrized in terms of analytical functions of suitably chosen internal coordinates. The potential energy terms can be expressed in atomic cartesian displacement coordinates as explained, for instance, in the book by Niketić and Rasmussen [13]. A force field parametrized in this way can be also be treated by MOLVIB. In this case the dynamical matrix is expressed directly in cartesian coordinates by a second-order expansion of the potential energy.

2.3 Special interactions.

As mentioned previously, overlay force field calculations are based on the transferability of force constants between similar atomic groups in the same or different molecules. To make such calculations meaningful, the interactions have to be chosen in a reasonable way.

This means that small and insignificant force constants, which cannot be well determined by least squares methods, should be omitted. How can we choose the interaction constants in a consistent way ?

Usually the interactions between neighbouring atomic groups are the most important ones. When the F-matrix is given in internal coordinates, this fact can be expressed as follows: "Interactions between coordinates having *no* common atoms are usually omitted".

A similar principle was already mentioned in the pioneering work by Schachtschneider and Snyder [18]. According to them, "no attempt was made to determine interaction constants between co-ordinates centered on carbon atoms further removed than second nearest neighbors".

The principles for choosing the significant interaction constants are not always explicitly stated. However, Abe and Krimm [1] give the following simple rules to choose the interactions:

a) Interactions between the bond stretching coordinates for a pair of bonds which share a common atom:

$V(r_1, r_2) = f(r_1, r_2)\Delta r_1 \Delta r_2$

b) Interaction between the bending coordinate for a bond angle and the stretching coordinate for a bond attached to the apex atom of the angle:

$V(r, \theta) = f(r, \theta)\Delta r \Delta\theta$

c) Interaction between the bending coordinates for a pair of bond angles:

$V(\theta_1, \theta_2) = f(\theta_1, \theta_2)\Delta\theta_1 \Delta\theta_2$

d) Interaction between the out-of-plane bending coordinates for a pair of trigonal planar groups, or between the bending coordinate for a bond angle and the out-of-plane bending coordinate for a planar group which share a common bond and/or a common apex atom;

$V(w_1, w_2) = f(w_1, w_2)\Delta w_1 \Delta w_2$

$V(\theta, w) = f(\theta, w)\Delta\theta \Delta w$

e) Interaction between the torsional coordinate for a bond and an out-of-plane bending coordinate for a group attached to the bond:

$V(w, t) = f(w, t)\Delta w \Delta t$

These are the definitions for the interaction force constants of the peptide force field developed by Krimm and his coworkers in the University of Michigan. The principles are, of course, quite general, and can be applied to any molecule containing planar groups. All of the special interactions discussed above can be generated by the programme MOLVIB.

To make it easier to define transferable force constants, one can apply certain general rules due to Hollenstein and Günthard [9]. They suggest a scheme of dividing the atoms of a molecule into subsets according to the following principles:
$T_0(s)$ denotes the set of atoms involved in the internal coordinate s,
$T_1(s)$ contains the atoms bonded to the atoms in $T_0(s)$,
...
$T_k(s)$ are the atoms that are bonded to the atoms in $T_{k-1}(s)$, and so forth.

Analogously $T_0(s, s')$ denotes the set of atoms involved in at least one of the coordinates s and s', i.e., $T_0(s, s') = T_0(s) \cup T_0(s')$.

A force constant connected with an internal coordinate s is then considered transferable, if the atom configurations of the corresponding sets $T_0(s)$, $T_1(s)$ and $T_2(s)$ in the molecules are the same, and the types and directions of the bonds connecting the sets $T_2(s)$ and $T_3(s)$ are similar. To simplify the force field, one may perhaps remove some of the conditions.

An interaction constant associated with the internal coordinates s and s' is considered transferable, if the atom configuration of the corresponding sets $T_0(s, s')$ and $T_1(s, s')$ are almost the same. This rule can also be made less strict by removing the condition on T_1.

2.4 Choice of internal coordinates in force field calculations

In overlay calculations, transferability of the force constants of similar atomic groups between different molecules is assumed. Although the force constants of one group may be somewhat affected by the neighbouring groups, the work of Snyder, Schachtschneider, Shimanouchi and others [17-19] have shown this assumption to be reasonably true.

These overlay calculations have been based on various types of force fields. Many years ago, a new force field concept was introduced by Shimanouchi and his coworkers [18]. This force field, which is called the local symmetry force field (LSFF), is based on the following principles: a) The symmetry coordinates for the atomic groups (CH_3, CH_2 ,. . .) are used as basic coordinates, and the force constants are expressed in terms of these coordinates. Force constants expressed in symmetry coordinates are usually not as much correlated as force constants expressed in internal coordinates, if the redundancies are properly removed. b) Only the interaction constants between *next*-neighbouring groups are taken into account, if there are no special reasons to include interactions between groups further away.

The definitions of the local symmetry coordinates are based on certain conventions. Thus the signs of all non-terminal groups in the molecule will be positive, if the molecule is viewed from left to right according to the molecular formula. This is to ensure that the interaction constants will have the proper signs.

Shimanouchi's local symmetry coordinates are very similar to the set of internal coordinates, called *natural coordinates* that were introduced by Pulay and his coworkers in 1979 [14] for ab initio vibrational studies. These coordinates are based on three principles: locality, local pseudosymmetry and elimination of redundancy. Locality implies (approximate) local symmetry. Pseudosymmetry around center atoms is used to define composite angle deformation coordinates. Special coordinates have been introduced for rings, relative torsion around a bond is described by the arithmetic mean of the torsion angles involved, and so on. Such coordinates have to be explicitly defined in the MOLVIB program, using a simple syntax for constructing linear combinations, as will be described later.

Many molecular organometallic complexes (metallocenes, e.g.) involve carbon rings, which interact with the metal atoms by translation or bending of the ring *as a whole.* As discussed by Doman et al. [7], these motions can be described by substituting the ring with a *dummy atom*, situated in the ring center. Force constants are then assigned to the metal-dummy stretches, metal-dummy bends, and so on. All of the forces associated with the dummy atom become, however, distributed among the atoms that define the dummy, leaving no residual forces on the dummy atom. This method of defining ligand coordinates has been implemented in MOLVIB (see sect. 3.3.1 and 3.3.1.2).

2.5 Optimization of force constants in MOLVIB.

As was previously shown, the eigenvalue equation of the dynamical matrix can be expressed as a set of linear equations:

$$\sum_{k=1}^{m}(\tilde{U}M^{-1/2}C^{(k)}M^{-1/2}U)_{ii}f_k = \lambda_i \quad (i = 1, 2, \ldots, 3N).$$

Provided that the orthogonal transformation U is known and that $m < 3N - 6$ (the number of non-zero eigenvalues in the general case) these equations can, in principle, be solved by the method of *linear* least squares.

However, the matrix U is usually *not* known, and consequently these equations have to be solved repeatedly by making approximations to the force constants that reproduce the observed frequencies with increasing accuracy. How can this be done in practice ?

The problem would be much simplified, if the above-mentioned equations could be used to relate the changes in the force constants to changes in the eigenvalues. Actually it is possible to approximate the above equation with the much simpler *linear* relation

$$\sum_{k=1}^{m}(\tilde{U}M^{-1/2}C^{(k)}M^{-1/2}U)_{ii}\Delta f_k \approx \Delta\lambda_i.$$

Strictly speaking, this approximation is valid *only* when the calculated and observed eigenvalues are "almost" coincident. However, it is commonly used also in the beginning of the iteration process, when the changes in the force constants may be quite large, and consequently can produce large changes in the eigenvalues and eigenvectors. As can be shown by practical calculations, it is usually more efficient to use some kind of *damped* least squares procedure instead of trying to use a more accurate non-linear relationship to compute the changes in the force constants. These methods will be discussed next.

In order to apply the method of least squares to the force constant calculations, we have to define a quadratic function (also called *objective function*) that is to be minimized with respect to the force constants. This *goodness-of-fit* function is defined as follows:

$$S(\lambda) = (\lambda - \lambda^0)^T W(\lambda - \lambda^0)$$

where λ^0 denotes a set of observed eigenvalues (including one or several molecules), λ is the corresponding calculated eigenvalue vector, and W is a diagonal weight matrix.

Sometimes the goodness-of-fit function is defined as a function of the normal frequencies ν_i, and the weights are applied to the frequencies. However, if the observed and calculated frequencies are close to each other, which is the case in the neighborhood of a minimum, the two ways of defining S will give essentially the same results, if the weights are chosen appropriately. From the relation between the normal frequencies and the eigenvalues (see p. 3), and the definition of S we find ($k = 1302.79$ cm^{-1})

$$S = k^{-4}\sum_i w_i[\nu_i^2 - (\nu_i^0)^2]^2 \approx 4k^{-4}\sum_i w_i(\nu_i^0)^2(\nu_i - \nu_i^0)^2,$$

which shows that the choice $w_i = (\lambda_i^0)^{-1}$ is approximately equivalent to giving all the frequencies *equal* weights.

Because the eigenvalues are related to the force constants, S will also be related to the force constants: $S(\lambda(f)) \equiv \bar{S}(f)$. The least squares solution can then be found from the condition that the gradient vector of S in the force constant space vanishes:

$$\nabla\bar{S}(f) = 0.$$

Assume now, that f' is a force constant vector close to the optimal value f. If the gradient $\nabla\bar{S}(f)$ is expanded in a Taylor's series around f, we get

$$\nabla\bar{S}(f) = \nabla\bar{S}(f') + \nabla\tilde{\nabla}\bar{S}(f')(f - f') + \ldots,$$

where $\nabla\tilde{\nabla}\bar{S}(f')$ denotes the Hessian of $\bar{S}(f')$.

Differentiating the eigenvalue equation of the dynamical matrix and assuming that the orthogonal transformation U is approximately independent of f (see, however, the comments at the beginning of this section) we find the elements of the Jacobian matrix J:

$$\frac{\partial \lambda_i}{\partial f_j} \approx (\tilde{U} M^{-1/2} C^{(j)} M^{-1/2} U)_{ii} = J_{ij},$$

or, in matrix notation, $\nabla\lambda = J$.

The gradient and Hessian of $\bar{S}(f')$ can be expressed as

$$\begin{aligned} \nabla\bar{S}(f') &= 2(\nabla\lambda(f'))^T W(\lambda(f') - \lambda^0) \\ \nabla\tilde{\nabla}\bar{S}(f') &= 2[(\nabla\lambda(f'))^T W \nabla\lambda(f') + (\nabla\tilde{\nabla}\lambda(f'))W(\lambda(f') - \lambda^0)]. \end{aligned}$$

If we now assume that the residuals have a random distribution, and that the second order derivatives of the eigenvalues with respect to the force constants do not vary much, the second order terms in the Hessian can be neglected. We then get the following approximate expressions for the gradient and the Hessian

$$\begin{aligned} \nabla\bar{S}(f') &\approx 2\tilde{J}W(\lambda' - \lambda^0) \\ \nabla\tilde{\nabla}\bar{S}(f') &\approx 2(\nabla\lambda(f'))^T W \nabla\lambda(f') \approx 2\tilde{J}WJ. \end{aligned}$$

This leads to a solution technique for the least squares equations known as the *Gauss–Newton* method. Obviously, it is strictly valid only close to a local minimum of S.

In this approximation the higher derivatives vanish, and the minimum condition for S gives:

$$\tilde{J}WJ(f - f') \approx -\tilde{J}W(\lambda' - \lambda^0).$$

The corrections to the force constants are then found by repeated application of the formula

$$\Delta f = -A^{-1}y,$$

where $A = \tilde{J}WJ$ and $y = \tilde{J}W(\lambda' - \lambda^0)$.

In the neighborhood of a local minimum of S, A is always positive semi-definite. However, if the matrix A is nearly singular ($\det A \approx 0$), the changes in the force constants may become too large and the iteration process diverges. This happens far from the minimum or when the force constants are strongly correlated (e.g., when the force constants are not independent or when the number of degrees of freedom is too small). As we already noted, the Gauss-Newton method is not valid far from the minimum. However, inclusion of the second order terms in the Hessian will usually only speed up the iteration process in the beginning.

In such cases, a *damped least squares method* should be used. This means that the last mentioned equation is replaced by

$$\Delta f = -A^* y,$$

where A^* is a "generalized inverse" of A. There are several methods for defining A^*.

One method, originally due to Levenberg [11] and later on modified by Marquardt [12], has been much used in force field calculations, and is implemented in MOLVIB (option MARQ). In this method, $A^* = (A+\beta B)^{-1}$, where β is a positive number and B is a diagonal matrix of the same order as A. In Levenberg's method, B is the identity matrix, whereas Marquardt chose the elements of B equal to the diagonal elements of A. By a suitable choice of β, $A + \beta B$ will become positive definite and the force constants calculated from the above equation will lead to a decrease in S. The strategy usually adopted is to increase β by a factor k_1 if S increases, but to *decrease* β by a factor k_2 if S decreases. Marquardt suggested $k_1 = 2$ and $k_2 = 10$. However, it seems that $k_2 = 10$ is too large in practical calculations, and therefore $k_2 = 3$ has been used in the MOLVIB implementation.

A very interesting method for the automatic determination of β has been suggested by Davies and Whitting [6], who express β in the form

$\beta = \tilde{y}B^{-1}y/S.$

This strategy is also implemented in MOLVIB (option MMARQ).

A generalized inverse of A can also be constructed by a *modified Gauss method* (MOLVIB option MGAUS). According to this method, the matrix A is first diagonalized by an orthogonal transformation V:

$\tilde{V}AV = \Gamma.$

Here Γ denotes the diagonal eigenvalue matrix defined by $\Gamma_{ij} = \gamma_i\delta_{ij}$. Obviously, A can then be expressed as

$A = V\Gamma\tilde{V},$

and the inverse matrix is given by the analogous formula

$A^{-1} = V\Gamma^{-1}\tilde{V}.$

However, if A is singular, or nearly singular, this equation cannot be used, because some of the eigenvalues may become zero or very small. A generalized inverse can then be defined by

$A^* = V\Gamma^*\tilde{V},$

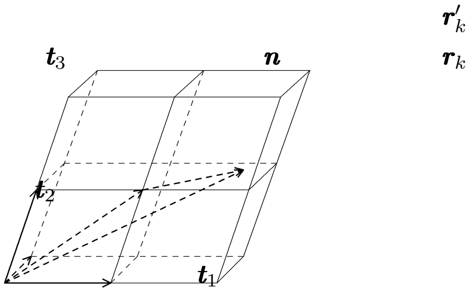
where Γ^* is a diagonal matrix, whose elements γ_i are related to the corresponding eigenvalues of A. A simple method to choose the elements γ_i has been found by the author [22], and has been implemented in MOLVIB. This method gives often rise to faster convergence than the Marquardt method if the force constants are correlated.

The calculation of the goodness-of-fit function S implies a proper assignment of the observed eigenvalues to the calculated normal modes. This can be achieved as follows. Assume that U_0 is an orthogonal matrix that diagonalizes the dynamical matrix for a certain molecule calculated with a given set of force constants f_0 (called a *reference point*), and that that U is another orthogonal matrix that diagonalizes the dynamical matrix for the same molecule calculated with another set of force constants f . The matrix $V = \tilde{U}_0U$ is then constructed and the element of largest modulus V_{ij_m} in each row is determined. If the index $j_m = i$, then the column vectors of U correspond exactly to those of U_0 (i.e., the assignment is *preserved*). If $j_m \neq i$, an interchange of the columns has occurred, and the new order is recorded. During the course of refinement, it may happen that the force constants change so much that the eigenvectors become badly mixed. Several elements in the same row of V can then be of the same order of magnitude, and it is possible, that the index j_m has been recorded previously. If this is the case, the second largest element in the row is determined, and the above procedure is repeated.

2.6 The theory of lattice vibrations.

A crystal can be considered as an infinite three-dimensional lattice, which is described by a primitive unit cell that generates the lattice by simple translations.

The translational symmetry means that the Hamiltonian of the crystal will be invariant with respect to a *translation vector* (or lattice vector) $\boldsymbol{n}$, defined by



$$\boldsymbol{n} = n_1\boldsymbol{t}_1 + n_2\boldsymbol{t}_2 + n_3\boldsymbol{t}_3,$$

where the n_i:s are arbitrary integers and the $\boldsymbol{t}_i$:s are three non-coplanar basis vectors of the unit cell. This vector indicates the origin of any cell with respect to a reference unit cell. The position of each atom k with respect to the origin of the cell is denoted $\boldsymbol{r}'_k$. Using the translation vector the position of this atom with respect to the reference cell can be expressed $\boldsymbol{r}_k = \boldsymbol{n} + \boldsymbol{r}'_k$ (see the figure).

To each set of basis vectors one can add a set of three *reciprocal vectors*, defined by the equation $\boldsymbol{t}_i \cdot \boldsymbol{u}_j = \delta_{ij}$. Naturally, if all $\boldsymbol{t}_i$:s are mutually orthogonal, they will coincide with the reciprocal vectors. Generally the vectors $\boldsymbol{u}_i$ are said to span the *reciprocal lattice.* A lattice translation vector in reciprocal space can be defined by

$$\boldsymbol{k} = h_1\boldsymbol{u}_1 + h_2\boldsymbol{u}_2 + h_3\boldsymbol{u}_3,$$

where the h_i:s are integers. Thus the scalar product

$$\boldsymbol{k} \cdot \boldsymbol{n} = k_1h_1 + k_2h_2 + k_3h_3$$

is also an integer.

To explore translational symmetry further let us consider an arbitrary function f of the lattice translation vector $\boldsymbol{n}$. Because of the translational symmetry, we must have

$$f(\boldsymbol{n}+\boldsymbol{n}') = f(\boldsymbol{n})$$

for all vectors $\boldsymbol{n}'$. Especially we will get

$$\mathrm{e}^{2\pi i\boldsymbol{k}\cdot\boldsymbol{n}} = \mathrm{e}^{2\pi i\boldsymbol{k}\cdot(\boldsymbol{n}+\boldsymbol{n}')} = \mathrm{e}^{2\pi i\boldsymbol{k}\cdot\boldsymbol{n}'}\mathrm{e}^{2\pi i\boldsymbol{k}\cdot\boldsymbol{n}},$$

where $\boldsymbol{k}$ is a wave vector (associated with *phonons*, the quanta of lattice-vibrational energy).

Summing over all the lattice vectors $\boldsymbol{n}$, we then get

$$\sum_{\boldsymbol{n}} \mathrm{e}^{2\pi i\boldsymbol{k}\cdot\boldsymbol{n}} = \mathrm{e}^{2\pi i\boldsymbol{k}\cdot\boldsymbol{n}'} \sum_{\boldsymbol{n}} \mathrm{e}^{2\pi i\boldsymbol{k}\cdot\boldsymbol{n}},$$

which shows that

$$\mathrm{e}^{2\pi i \boldsymbol{k} \cdot \boldsymbol{n}'} = 1$$

and that $\boldsymbol{k} \cdot \boldsymbol{n}' = q$ (q is any integer). Because $\boldsymbol{n}'$ is assumed to be a translation vector, $\boldsymbol{k}$ must be a reciprocal translation vector (with integral components).

The vector $\boldsymbol{k} = \boldsymbol{0}$ (corresponds to optically active vibrations) is, of course, also a valid reciprocal translation vector. If $\boldsymbol{k} = \boldsymbol{0}$ then

$$\sum_{\boldsymbol{n}} \mathrm{e}^{2\pi i \boldsymbol{k} \cdot \boldsymbol{n}} = \sum_{\boldsymbol{n}} \mathrm{e}^{0} = N,$$

where N denotes the number of unit cells in the crystal. This formula can also be interpreted as the orthogonality condition of two plane waves with wave vectors $\boldsymbol{k}$ and $\boldsymbol{k}'$:

$$\frac{1}{N} \sum_{\boldsymbol{n}} \mathrm{e}^{2\pi i \boldsymbol{k} \cdot \boldsymbol{n}} \mathrm{e}^{-2\pi i \boldsymbol{k}' \cdot \boldsymbol{n}} = \frac{1}{N} \mathrm{e}^{2\pi i (\boldsymbol{k} - \boldsymbol{k}') \cdot \boldsymbol{n}} = \delta(\boldsymbol{k} - \boldsymbol{k}'),$$

where the δ-function has the special meaning that it is equal to zero unless $\boldsymbol{k} - \boldsymbol{k}'$ is a reciprocal lattice vector (i.e., expressible as $\sum_i h_i \boldsymbol{u}_i$).

An arbitrary function of a lattice vector $\boldsymbol{n}$ can be expanded in plane waves:

$$f(\boldsymbol{n}) = N^{-1/2} \int g(\boldsymbol{k}) \mathrm{e}^{-2\pi i \boldsymbol{k} \cdot \boldsymbol{n}} d\boldsymbol{k}.$$

Using the orthogonality condition, one can easily find the inverse transformation:

$$\begin{aligned} \sum_{\boldsymbol{n}} f(\boldsymbol{n}) \mathrm{e}^{2\pi i \boldsymbol{k} \cdot \boldsymbol{n}} &= N^{-1/2} \sum_{\boldsymbol{n}} \int g(\boldsymbol{k}') \mathrm{e}^{-2\pi i \boldsymbol{k}' \cdot \boldsymbol{n}} \mathrm{e}^{2\pi i \boldsymbol{k} \cdot \boldsymbol{n}} d\boldsymbol{k}' \\ &= N^{-1/2} \int d\boldsymbol{k}' g(\boldsymbol{k}') \sum_{\boldsymbol{n}} \mathrm{e}^{2\pi i (\boldsymbol{k} - \boldsymbol{k}') \cdot \boldsymbol{n}} \\ &= N^{1/2} \int d\boldsymbol{k}' g(\boldsymbol{k}') \delta(\boldsymbol{k} - \boldsymbol{k}'), \end{aligned}$$

so that

$$g(\boldsymbol{k}) = N^{-1/2} \sum_{\boldsymbol{n}} f(\boldsymbol{n}) \mathrm{e}^{2\pi i \boldsymbol{k} \cdot \boldsymbol{n}}.$$

Assume now that there are σ molecules in each unit cell. Then, using the translational symmetry, the position of each molecule in the crystal can be uniquely specified by giving the lattice vector $\boldsymbol{n}$ and the number i ($i = 1, 2, \ldots, \sigma$) that determines the position of the molecule in the unit cell, i.e. the *site*.

To simplify matters we shall now assume that the vibrations of the N' atoms in the j:th unit cell can be alternatively described by $3N$ cartesian displacement coordinates collected in a vector $\boldsymbol{x}_j$, or by $3N$ independent internal coordinates collected in a vector $\boldsymbol{s}_j$. Let $\boldsymbol{n}_j$ denote the translation vector of this unit cell.

Instead of using the ordinary cartesian and internal displacement vectors, we shall introduce the so-called *phonon coordinate vectors*:

$$\boldsymbol{X}(\boldsymbol{k}) = N^{-1/2}\sum_j \boldsymbol{x}_j \mathrm{e}^{2\pi i \boldsymbol{k}\cdot\boldsymbol{n}_j}$$
$$\boldsymbol{S}(\boldsymbol{k}) = N^{-1/2}\sum_j \boldsymbol{s}_j \mathrm{e}^{2\pi i \boldsymbol{k}\cdot\boldsymbol{n}_j}$$

The next problem to be solved is to find the relation between the internal and the cartesian coordinates. Using Wilson's method the internal coordinate vector $\boldsymbol{s}_j$ can be expressed as

$$\begin{aligned}\boldsymbol{s}_j &= B_{-m}\boldsymbol{x}_{j-m} + \ldots + B_{-1}\boldsymbol{x}_{j-1} + B_0\boldsymbol{x}_j + B_{+1}\boldsymbol{x}_{j+1} + \ldots + B_m\boldsymbol{x}_{j+m}\\ &= \sum_{l=-m}^{m} B_l\boldsymbol{x}_{j+l},\end{aligned}$$

where $\boldsymbol{x}_{j+l}$ denotes the cartesian vector of a unit cell, l units apart from the n-th cell. Transforming this equation to phonon coordinates, we find

$$\begin{aligned}\boldsymbol{S}(\boldsymbol{k}) &= N^{-1/2}\sum_j\sum_l B_l\boldsymbol{x}_{j+l}\mathrm{e}^{2\pi i \boldsymbol{k}\cdot\boldsymbol{n}_j}\\ &= N^{-1/2}\sum_l B_l \mathrm{e}^{-2\pi i \boldsymbol{k}\cdot\boldsymbol{n}_l}\sum_j \boldsymbol{x}_{j+l}\mathrm{e}^{2\pi i \boldsymbol{k}\cdot(\boldsymbol{n}_j+\boldsymbol{n}_l)}\\ &= \sum_l B_l \mathrm{e}^{-2\pi i \boldsymbol{k}\cdot\boldsymbol{n}_l}\boldsymbol{X}(\boldsymbol{k}).\end{aligned}$$

This means that the relation can be written as

$$\boldsymbol{S}(\boldsymbol{k}) = B(\boldsymbol{k})\boldsymbol{X}(\boldsymbol{k}),$$

where

$$B(\boldsymbol{k}) = \sum_l B_l \mathrm{e}^{-2\pi i \boldsymbol{k}\cdot\boldsymbol{n}_l}.$$

On the other hand, the potential energy of the crystal is given by

$$2V = \sum_j\sum_{j'} \boldsymbol{s}_j^\dagger F_{jj'}\boldsymbol{s}_{j'}.$$

However, the translational symmetry requires that $F_{jj'}$ should only depend on the difference $|j-j'| = l$, thus $F_{jj'} = F_l$. From this follows

$$2V = \sum_j\sum_l \boldsymbol{s}_j^\dagger F_l\boldsymbol{s}_{j+l}.$$

By insertion of the inverse transformation

$$\boldsymbol{s}_j = N^{-1/2}\int \boldsymbol{S}(\boldsymbol{k})\mathrm{e}^{-2\pi i \boldsymbol{k}\cdot\boldsymbol{n}_j}\,d\boldsymbol{k}$$

we find

$$\begin{aligned}2V &= N^{-1}\sum_j\sum_l\int\int \boldsymbol{S}^{\dagger}(\boldsymbol{k})\mathrm{e}^{2\pi i\boldsymbol{k}\cdot\boldsymbol{n}_j}F_l\boldsymbol{S}(\boldsymbol{k})\mathrm{e}^{-2\pi i\boldsymbol{k}'\cdot(\boldsymbol{n}_j+\boldsymbol{n}_l)}d\boldsymbol{k}d\boldsymbol{k}'\\ &= \sum_l\int\int \boldsymbol{S}^{\dagger}(\boldsymbol{k})F_l\mathrm{e}^{-2\pi i\boldsymbol{k}'\cdot\boldsymbol{n}_l}\boldsymbol{S}(\boldsymbol{k}')\delta(\boldsymbol{k}-\boldsymbol{k}')d\boldsymbol{k}d\boldsymbol{k}'\\ &= \int \boldsymbol{S}^{\dagger}(\boldsymbol{k})\sum_l F_l\mathrm{e}^{-2\pi i\boldsymbol{k}\cdot\boldsymbol{n}_l}\boldsymbol{S}(\boldsymbol{k})d\boldsymbol{k}.\end{aligned}$$

Introducing the notation

$$F(\boldsymbol{k}) = \sum_l F_l\mathrm{e}^{-2\pi i\boldsymbol{k}\cdot\boldsymbol{n}_l},$$

the expression for the potential energy can be written in the simple form

$$2V = \int \boldsymbol{S}^{\dagger}(\boldsymbol{k})F(\boldsymbol{k})\boldsymbol{S}(\boldsymbol{k})d\boldsymbol{k}.$$

In a similar way one can show that the F-matrix expressed in cartesian displacement coordinates $F_x(\boldsymbol{k})$ is given by

$$F_x(\boldsymbol{k}) = B^{\dagger}(\boldsymbol{k})F(\boldsymbol{k})B(\boldsymbol{k})$$

and that the dynamical matrix $D(\boldsymbol{k})$ can be expressed as

$$\begin{aligned}D(\boldsymbol{k}) &= M^{-1/2}F_x(\boldsymbol{k})M^{-1/2}\\ &= M^{-1/2}B^{\dagger}(\boldsymbol{k})F(\boldsymbol{k})B(\boldsymbol{k})M^{-1/2}.\end{aligned}$$

2.7 The calculation of crystal normal coordinates.

In general, the dynamical matrix $D(\boldsymbol{k})$ will be a complex matrix of order n. The diagonalization of this matrix does not, however, present any problem, because it can be shown to be equivalent with a real matrix of order $2n$.

Expressing the matrix D in the form $D = D_r + iD_s$ and the eigenvector x as $x_r + ix_s$ we find that the eigenvalue equation may be written in the form

$$(D_r + iD_s)(x_r + ix_s) = \lambda(x_r + ix_s)$$

which leads to the following two equations (one for the real part and one for the imaginary part):

$$\begin{aligned}&D_rx_r - D_sx_s = \lambda x_r\\ &D_sx_r + D_rx_s = \lambda x_s.\end{aligned}$$

These equations can be combined into a single matrix eqaution:

$$\begin{pmatrix} D_r & -D_s\\ D_s & D_r\end{pmatrix}\begin{pmatrix} x_r\\ x_s\end{pmatrix} = \lambda\begin{pmatrix} x_r\\ x_s\end{pmatrix},$$

which shows that the real matrix

$$\begin{pmatrix} D_r & -D_s \\ D_s & D_r \end{pmatrix}$$

has the same eigenvalues and eigenvectors as the complex matrix D.

Writing the matrix D as

$$D(\boldsymbol{k}) = \sum_l D_l \mathrm{e}^{-i\boldsymbol{k}\cdot\boldsymbol{n}_l} = \sum_l D_l \cos(\boldsymbol{k}\cdot\boldsymbol{n}_l) + i\sum_l D_l \sin(\boldsymbol{k}\cdot\boldsymbol{n}_l)$$

we find that the real matrix corresponding to $D(\boldsymbol{k})$ is

$$\begin{pmatrix} \sum_l D_l \cos(\boldsymbol{k}\cdot\boldsymbol{n}_l) & -\sum_l D_l \sin(\boldsymbol{k}\cdot\boldsymbol{n}_l) \\ \sum_l D_l \sin(\boldsymbol{k}\cdot\boldsymbol{n}_l) & \sum_l D_l \cos(\boldsymbol{k}\cdot\boldsymbol{n}_l) \end{pmatrix}.$$

A very important case is when $\boldsymbol{k} = \mathbf{0}$, which corresponds to the *optically* active vibrations. In this case the D-matrix is *real*. Thus the above mentioned augmentation of the matrix need not be done, which saves computer memory space and computation time.

In this case the D-matrix may be factorized into blocks which are associated with symmetry species. This can be found in the following way. Assume that each unit cell can be subdivided into m asymmetric units. Then the structure of the $B(\mathbf{0})$ matrix can be described as follows:

$$\begin{array}{llll}
 & \tilde{X}_1(\mathbf{0}) & \tilde{X}_2(\mathbf{0})\ldots & \tilde{X}_m(\mathbf{0}) \\
S_1(\mathbf{0}) & B_{11}(\mathbf{0}) & B_{12}(\mathbf{0})\ldots & B_{1m}(\mathbf{0}) \\
S_2(\mathbf{0}) & B_{21}(\mathbf{0}) & B_{22}(\mathbf{0})\ldots & B_{2m}(\mathbf{0}) \\
\ldots & \ldots & \ldots\ldots & \ldots \\
S_m(\mathbf{0}) & B_{m1}(\mathbf{0}) & B_{m2}(\mathbf{0})\ldots & B_{mm}(\mathbf{0})
\end{array}$$

$B_{ij}(\mathbf{0})$ denotes the B-matrix which connects the internal coordinates $S_i(\mathbf{0})$ of the i:th asymmetric unit with the cartesian coordinates $X_j(\mathbf{0})$ of the j:th asymmetric unit.

Let T_p denote a 3×3 matrix whic moves the p:th asymmetric unit into the same orientation as the first asymmetric unit. T_p can obviously be written in the form of a rotation matrix. Assuming that there are m' atoms in the asymmetric unit, we can define a transformation matrix U_p which transforms the cartesian displacement coordinates of the first asymmetric unit into those of the p:th asymmetric unit:

$$U_p X_1 = X_p.$$

This matrix can be expressed as as the direct product of a unit matrix $I_{m'}$ of order m' and the matrix T_p:

$$U_p = I_{m'} \otimes T_p = \begin{pmatrix} T_p & 0 & 0 & \ldots & 0 \\ 0 & T_p & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & T_p \end{pmatrix}.$$

The B-matrix associated with a certain symmetry species Γ can then be expressed as

$$B(\Gamma) = \sum_{p=1}^{m} c_p(\Gamma) B_{1p}(\mathbf{0}) \tilde{U}_p,$$

where $c_p(\Gamma)$ denotes a coefficient found from a point group character table.

The F-matrix $F_s(\mathbf{0})$ has the corresponding structure:

$$\begin{array}{llll}
 & \tilde{S}_1(\mathbf{0}) & \tilde{S}_2(\mathbf{0})\ldots & \tilde{S}_m(\mathbf{0}) \\
S_1(\mathbf{0}) & F_{11}(\mathbf{0}) & F_{12}(\mathbf{0})\ldots & F_{1m}(\mathbf{0}) \\
S_2(\mathbf{0}) & F_{21}(\mathbf{0}) & F_{22}(\mathbf{0})\ldots & F_{2m}(\mathbf{0}) \\
\ldots & \ldots & \ldots\ldots & \ldots \\
S_m(\mathbf{0}) & F_{m1}(\mathbf{0}) & F_{m2}(\mathbf{0})\ldots & F_{mm}(\mathbf{0})
\end{array}$$

Here the diagonal terms $F_{ii}(\mathbf{0})$ arise from the intramolecular terms of the potential energy in the i:th asymmetric unit, and the off-diagonal terms $F_{ij}(\mathbf{0})$, which are symmetric, are due to the interactions between the i:th and j:th asymmetric units.

The potential energy matrix with respect to the symmetry species Γ can be expressed as

$$F(\Gamma) = \sum_{p=1}^{m} c_p(\Gamma) F_{1p}(\mathbf{0}),$$

and the corresponding dynamical matrix as

$$D(\Gamma) = M^{-1/2} \tilde{B}(\Gamma) F(\Gamma) B(\Gamma) M^{-1/2}.$$

A similar description can be found in the paper by Kobayashi [10].

2.8 Scaling factor calculations in MOLVIB

In a scaling factor calculation by MOLVIB, the scaling factors are treated as ordinary force constants. They are thus calculated from a least squares fit of the calculated and observed frequencies, as described previously. These scaling factors can also be applied to a group of similar force constants.

To perform the scaling factor calculations, the program needs the atomic coordinates, and the cartesian force constants from an *ab initio* calculation. An auxiliary program (`rdarch`) that extracts these data from the archive part of a Gaussian log file has been written. In addition, it can also extract the dipole derivatives and the polarization derivatives.

MOLVIB will convert the Gaussian force constants, which are expressed in atomic units, into those used by MOLVIB. Pulay has used what he calls a 'reference' geometry, which is close to the experimental geometry, but presently MOLVIB can only calculate frequencies for a molecule in an equilibrium geometry (as the Gaussian program does). It is always good to start with an initial calculation, where all the scaling factors have been set to 1, and check if one can reproduce the frequencies calculated by the ab initio program.

In the scaling factor calculations, three methods can be used, as defined by the value of the option `SCALF`. Two of them are based on Pulay's method [15], according to which the internal force constants f_{ij} are transformed according to

$$f'_{ij} = \sqrt{S_i S_j} f_{ij},$$

(*geometric mean scaling*) where S_i denotes a scaling constant (usually less than 1).

A strict application of this method would mean that the non-diagonal terms in the potential energy will depend nonlinearly on the scaling factors:

$$2V = \sum_i S_i f_{ii} q_i q_i + \sum_i \sum_{j \neq i} \sqrt{S_i S_j} f_{ij} q_i q_j$$

In the simplest method (`SCALF = 1`), the factor $\sqrt{s_i s_j}$ that occurs in front of the non-diagonal force constant is therefore fixed during the iteration, which means that the calculation has to be repeated. In this case the frequency fit usually converges in four or five iterations, and often just a few repetitions are necessary. The initial values for the scaling factors are set to 1.

One can also use a variant of this method (`SCALF=2`), where the scaling factors associated with the non-diagonal force constants are updated during each iteration, but this produces usually a large number of non-zero C–matrix elements. On the other hand, the calculation needs not to be repeated, and is thus easier to use than the first method.

Lastly, it is also possible to use individual scaling factors for the non-diagonal force constants (`SCALF=3`). In this case, scaling factors should be associated both with diagonal and non-diagonal terms. Similar ideas have been proposed by Blom and Altona [3]. As is easy to understand, one should not use too many different scaling factors in this case, but instead group similar factors together, so that the total number of scaling factors is not more than about 10. Too many scaling factors can also give rise to large correlations.

Setting the initial values of the scaling factors to 1 usually leads to fast convergence, and the problems discussed in connection with ordinary force constant calculations (correlation, for instance) are not so important in scaling factor calculations, as long as the number of them is small. The Gauss-Newton method seems also to be quite adequate (cf. ref. [15b]), especially when it is combined with some kind of damping (modified Gauss or Marquardt). A more complicated algorithm is described in the paper by Allen [2].

The internal force constants, which are needed in the scaling, are found by the following coordinate transformation. According to definition, the internal coordinates of a molecule can be expressed in terms of cartesian coordinates as

$$s = Bx.$$

Usually the number of internal coordinates (m) is not equal to the number of cartesian coordinates (n). Thus the B–matrix is rectangular, and cannot be inverted in the usual way. If B^+ denotes a *pseudo-inverse* of the B-matrix:

$$B^+ = (B'B)^{-1}B',$$

we can write

$$x = B^+ s,$$

and the internal force constant matrix can then be found from a similarity transformation of the *ab initio* force constant matrix (the Hessian):

$$F_s = B'^+ F_x B^+.$$

Usually the scaling factors are applied to force constants expressed in local symmetry coordinates (Pulay uses 'natural' coordinates [14], which are similar), but the choice of the coordinates is not very crucial, provided that they are independent and form a complete set (redundant coordinates omitted). Observe, that we actually scale the internal force constants, not the cartesian force constants (Pulay has explained this in some detail in his papers). Using the PED option of the MOLVIB program, it is possible to get a list of the scaled internal force constants, and also the new cartesian force constants after scaling of the internal ones.

The program needs, of course, also the observed frequencies of the considered molecules (usually also some deuterated species are included, if frequencies are available). Ideally one should use harmonic frequencies, since the ab initio calculations are assumed to converge towards the harmonic frequencies. But if we wish to apply the scaled force field to some larger molecules for which no harmonic frequencies are available, the observed frequencies can be used. Pulay has remarked that in this case the scaling factors will also include some anharmonic effects, but such scaling factors may still be transferable to similar molecules. Of course, Fermi resonance causes a problem. This is the reason why the CH stretching frequencies, for instance, never are very well reproduced.

2.9 Intensity calculations in MOLVIB

The integrated molar absorption coefficient A_k of the k:th vibrational mode Q_k can be expressed as

$$A_k = \frac{1}{4\pi\epsilon_0}\frac{N_A\pi}{3c^2}\sum_g\left|\frac{\partial\mu}{\partial Q_k}\right|^2 = 974.86277\sum_g\left|\frac{\partial\mu}{\partial Q_k}\right|^2 \qquad \text{[km/mol]}$$

where the dipole derivatives are expressed in units of $eu^{-1/2}$ (summation is over the components of degenerate vibrations).

The components of the derivatives $\frac{\partial\mu}{\partial Q_k}$ are found from [24]

$$\frac{\partial\mu_i}{\partial Q_k} = \sum_\beta \frac{\partial\mu_i}{\partial x_\beta}\frac{u_{\beta,k}}{\sqrt{m_\beta}}$$

where $\frac{\partial\mu_\alpha}{\partial x_\beta}$ are the cartesian dipole derivatives, obtained from the Gaussian archive file (no final normalization is needed, see footnote 3 in ref. [16]). In the archive file, the dipole derivatives are listed in such an order that the components w.r.t. the space-fixed coordinates are changing most rapidly. The transition moments are also calculated from the dipole derivatives w.r.t. the normal coordinates.

In a similar way the polarizability derivatives w.r.t. the normal coordinates Q_k are found from the cartesian polarizability derivatives [25]:

$$\frac{\partial\alpha_{ij}}{\partial Q_k} = \sum_\beta \frac{\partial\alpha_{ij}}{\partial x_\beta}\frac{u_{\beta,k}}{\sqrt{m_\beta}}.$$

The mean polarisability is calculated from

$$\overline{\alpha} = \frac{1}{3}\sum_k\left(\frac{\partial\alpha_{11}}{\partial Q_k} + \frac{\partial\alpha_{22}}{\partial Q_k} + \frac{\partial\alpha_{33}}{\partial Q_k}\right),$$

and the anisotropy is found from

$$\gamma^2 = \frac{1}{2}\left\{\left(\frac{\partial\alpha_{11}}{\partial Q_k} - \frac{\partial\alpha_{22}}{\partial Q_k}\right)^2 + \left(\frac{\partial\alpha_{11}}{\partial Q_k} - \frac{\partial\alpha_{33}}{Q_k}\right)^2 + \left(\frac{\partial\alpha_{22}}{\partial Q_k} - \frac{\partial\alpha_{33}}{\partial Q_k}\right)^2 \right.$$
$$\left. +6\left[\left(\frac{\partial\alpha_{12}}{\partial Q_k}\right)^2 + \left(\frac{\partial\alpha_{13}}{\partial Q_k}\right)^2 + \left(\frac{\partial\alpha_{23}}{\partial Q_k}\right)^2\right]\right\}$$

The Raman activity can then be expressed as

$$\left(\frac{a_0}{\text{Å}}\right)^4 (45\overline{\alpha}^2 + 7\gamma^2) = 0.0784163 \cdot (45\overline{\alpha}^2 + 7\gamma^2),$$

and the depolarization degree as

$$\rho = \frac{3\gamma^2}{45\overline{\alpha}^2 + 4\gamma^2}$$

MOLVIB expresses the infrared and Raman intensities in the same units as the Gaussian program.

References

1. Y. Abe and S. Krimm, Biopolymers **11**, 1817-1839 (1972)
2. W.D. Allen, A.G. Császár, and D.A. Horner, J. Am. Soc. **114**, 6834 (1992)
3. C.E. Blom and C. Altona, Mol. Phys. **31**, 1377 (1976)
4. N. Bjerrum, Verh. d. Deutsch. Phys. Ges. **15**, 737 (1914)
5. S. Califano, Vibrational States, J. Wiley & Sons, London (1976)
6. M. Davies and I.J. Whitting, p. 191 in "Numerical Methods for Non-linear Optimization" (ed. F.A. Lootsma), Academic Press (1972)
7. T.N. Doman, C.R. Landis, and B. Bosnich, J. Am. Chem. Soc. **114**, 7264 (1992)
8. W.D. Gwinn, J. Chem. Phys. **55**, 477 (1971); H.L. Sellers, L.B. Sims, L. Schäfer, and D.E. Lewis, J. Mol. Struct. **41**, 149 (1977)
9. H. Hollenstein and Hs.H. Günthard, J. Mol. Spectrosc. **84**, 457 (1980)
10. M. Kobayashi, J. Chem. Phys. **70**, 4797 (1979).
11. K. Levenberg, Quart. Appl. Math. **2**, 164 (1944)
12. D.W. Marquardt, SIAM J. **11**, 431 (1963)
13. S.R. Niketić and Kj. Rasmussen, The Consistent Force Field: A Documentation, Lecture Notes in Chemistry, Vol. 3, Springer-Verlag, Heidelberg (1977)
14. P. Pulay, G. Fogarasi, F. Pang, and J.E. Boggs, J. Am. Chem. Soc. **101**, 2550 (1979); G. Fogarasi, X. Zhou, P.W. Taylor, and P. Pulay, J. Am. Soc. **114**, 8191 (1992)
15. P. Pulay, G. Fogarasi, G. Pongor, J.E. Boggs, and A. Vargha, J. Am. Chem. Soc. **105**, 7037 (1983); J. Baker, A.A. Jarzecki, and P. Pulay, J. Phys. Chem. **A 102**, 1412 (1998)
16. L.J. Schaad, C.S. Ewig, B.A. Hess, Jr., and D. Michalska, J. Chem. Phys. **83**, 5348 (1985)
17. J.H. Schachtschneider and R.G. Snyder, Spectrochim. Acta **19**, 117 (1963)
18. T. Shimanouchi, The Molecular Force Field, Ch. 6 in "Physical Chemistry: An Advanced Treatise", ed. H. Eyring, D. Henderson, and W. Jost (Vol. IV), Academic Press, New York (1970)
19. R.G. Snyder and J.H. Schachtschneider, Spectrochim. Acta **21**, 169 (1965)
20. T. Sundius, Comment. Phys.-Math. **47**, 1 (1977)
21. T. Sundius, J. Mol. Struct. **218**, 321 (1990)
22. T. Sundius, J. Mol. Spectrosc. **82**, 138 (1980)
23. E.B. Wilson, J.C. Decius and P.C. Cross, Molecular Vibrations, Mc Graw-Hill, New York (1955) (reprint: Dover, 1980)
24. D.J. Swanton, G.B. Bacskay, and N.S. Hush, Chem. Phys. **82**, 303 (1983)
25. D.J. Swanton, G.B. Bacskay, and N.S. Hush, Chem. Phys. **83**, 69 (1984)

3 Program manual

Beginning with version 6.0 of the program, it is possible to change several of the array dimensions and create larger (or smaller) executables. These dimensions are defined by PARAMETER statements in five INCLUDE files. Their default values are listed in the following table:

File name	Parameter	Value	Description
MOLATM.INC	MAXAT	50	Max. number of atoms (including crystal atoms)
MOLDAT.INC	MAXDAT	1000	Max. number of frequencies included in fit
MOLGEO.INC	MAXCOR	200	Max. number of (dependent) internal coordinates
MOLMAT.INC	NCMAX	100000	Max. number of cartesian C-matrix elements
		15000	in the MS Fortran version
	MAXDIM	150	Max. number of cartesian (or int.) coordinates
		100	in the MS Fortran version
MOLPAR.INC	NPMAX	100	Max. number of potential parameters
	INTMAX	1000	Max. number of defined force field interactions

The following disk files are used by the program:

Unit 9: A scratch file used by EVFIT and PED.

Unit 10: File `cmatrix` for storing the C-matrices.

Unit 11: File `bmatrix` for storing the B-matrices and geometrical information.

Unit 12: Scratch file used for the calculation of the B-matrix (MOLIN), the parameter error correlation matrix (MOLFIT), and the F-matrix (PED).

Unit 13: Scratch file used in the calculation of the C-matrices (MOLIN), and to store the symmetry transformation matrices (PED).

Unit 14: File `umatrix` for storing the eigenvectors and eigenvalues of the dynamical matrices.

Unit 15: File `vibpot` for storing force constant values and definitions.

Unit 16: File `cartfc` for storing cartesian force constants for scaling factor calculations (if SCALF > 1).

Unit 20: File `logfile` for storing logging information.

Unit 21: File `pedout` (real name is determined by the run script) for storing potential energy distributions.

Data is normally read from standard input (unit 5). In addition, a file may be specified for input of atomic coordinates and internal coordinates (control parameter: `DATFIL`). The unit number can be chosen by input.

Most of the printout from the program is directed to standard output (unit 6). However, the P.E.D. matrices generated by PED (using the control parameter `FILPED`) will be written to the file `pedout` on unit 21. Before a MOLVIB run is finished, logging information is written to the file `logfile` on unit 20.

For input, a free-format alphanumeric code is used, which will be described in the following.

3.0 Installation of the program

The program package contains detailed instructions for installation on different systems. Here some of the points are being discussed.

On a Unix system, MOLVIB can be installed by compiling all the subroutines with a Fortran (77 or 90) compiler and storing them in an object library `libmolvib.a`. A script file `mvblib` shows how this is done (for the f77 compiler).

Then the main program `molvib.for` is compiled and linked together with the library to create an executable `molvib` (see the script file `mvblink`). If the free compiler `g77` is used (in Linux, for instance) on should remember to use the option `-ffloat-store` when the main program `molvib.for` is compiled. To run the program, a script file `mvb` is used.

On a PC one can install the program under DOS (or Windows, under the command prompt) in similar way by first compiling the subroutines and then creating an object library `MOLVIB.LIB`. The file MVBLIB.BAT shows how this is done using the MS FORTRAN compiler.

The main program `MOLVIB.FOR` is then linked with the library to create an executable `MOLVIB.EXE` (see `MVBLINK.BAT`). A batch file `MVB.BAT` is provided for running the program.

Older versions of the MS FORTRAN compiler cannot use extended memory, and the program dimensions must therefore be small enough, so that the program runs in primary memory (cf. the table in the beginning of this chapter). However, using the Windows version of `g77` the memory restrictions can be eliminated, and it is then possible to use the same dimensions as in the Unix version of the program. Precompiled executables (both for MS FORTRAN and G77) have been provided in the program package.

A few of the routines are system dependent. These are `DATUM.FOR`, `TIME.FOR` and `MVBEND.FOR`. The Unix Fortran and MS FORTRAN versions of the main program are also slightly different. Using the VAX versions of these routines, it is also possible to run MOLVIB under VMS.

3.1 General Structure of the Input Data

1) A title line. The first column should contain a %-sign (the %-sign will be omitted from the output). Input lines are skipped until a title line is found.

2) Line(s) of input with values of the control parameters (read by the main program).

3) Lines of data (read by the subroutines, in the order they are called).

Data for several calculations may be added after each other, provided that the first line is a title line (see above).

3.2 Control Parameters

Twenty-two control parameters are presently recognized by the program :

`NRCALC`, `NBTYP`, `NMOLE`, `BMAT`, `BTRANS`, `PRINT`, `DATFIL`, `NEWPAR`, `NRITER`, `WEIGHT`, `MARQ`, `MMARQ`, `GAUSS`, `MGAUSS`, `SAVASS`, `PED`, `FILPED`, `UNITS`, `JACOB`, `SCALF`, `ZMAT` and `INTENS`.

The value of a parameter is specified by setting it equal to an integer number. Only the first three characters in a parameter name are necessary.

Example : `NRITER = 10`.

If the value is 1, only the name of the parameter need to be specified. The parameters and their values are listed in arbitrary order separated by commas. The end of the list is indicated by a semicolon (;). An omitted parameter is assumed to have the value 0 (the default value).

Example of a list of parameters:

```
NMOLE = 5, NBTYP =2, MGAUSS ,WEIGHT;
```

In the output, the parameters will listed together with their values. These parameters will be described in the following:

`NRCALC = 0` : No calculation (call of MOLFIT) performed. Can be used for testing, or when the B- and C-matrices are constructed separately, and saved on disk files for subsequent calculations. Previously this was sometimes necessary in order to save computing time.

`>0` : n calculations are performed (MOLFIT called n times).

`NBTYP = 0` : Urey-Bradley non-bonded (geminal) interactions are expressed in the harmonic approximation as $f(\Delta r)^2 + 2f'r\Delta r$, where f and f' are two independently variable force constants, and Δr is the geminal displacement coordinate (cf. sect. 2.2).

`= 1` : The ratio between f' and f is assumed constant (c), so that the interaction can be expressed as $f[(\Delta r)^2 + 2cr\Delta r]$. In this case the initial value of f' is substituted by c in the list of force constant definitions input to MOLIN.

`> 1` : Non-bonded interactions are treated as in the CFF-program (ref. 13) by computing the second-order derivatives of the potential energy terms. All the standard types of bonded diagonal terms (stretch, bend, out-of-plane bend, and torsion) can be treated by the program. For the non-bonded interactions Lennard-Jones (L-J) functions plus Coulomb terms are assumed.

`= 2` : The L-J function is assumed to be of the form: $a(r/2)^{-12} - b(r/2)^{-6}$

`= 3` : The L-J function is assumed to be of the form: $a[(r/r_0)^{-12} - 2(r/r_0)^{-6}]$, where r_0 is fixed.

`NMOLE = 0` : Files 10 and 11 are assumed to exist (The B- and C–matrices have been constructed separately, so MOLIN needs not to be called).

`=` n : Input data for n molecules is considered, and the files will be constructed.

`BMAT = 0` : The B-matrix will be generated by the program.

`= 1` : The B-matrix is given on input.

`BTRANS = 0` : No cartesian basis transformation.

`1` : A new basis is constructed from the eigenvectors of $M^{-1/2}B'BM^{-1/2}$. This usually leads to a smaller number of non-zero cartesian C-matrix elements, and can therefore sometimes be useful (*NB:* the molecular geometry should be fully three-dimensional).

`PRINT` =−2 : Smallest possible printout (output given only from MOLFIT and PED).

=−1 : No molecular input data will be printed, i.e. no output from MOLIN.

`= 0` : The non-zero elements of the B- and C-matrices will not be printed, and the eigenvectors from the diagonalization will not be printed in cartesian coordinates.

`= 1` : The non-zero elements of the B-matrix and the C-matrices (expressed in internal coordinates) will be printed. If `NBTYP` is greater than 1, the C-matrices are calculated directly in cartesian coordinates, and thus only the B-matrices will be printed.

`= 2` : The non-zero elements of the C-matrices (expressed in cartesian coordinates) will be printed.

`DATFIL = 0` : The molecular data are read from standard input (unit number 5).

`=` n : The molecular data are input from a file on unit n (16, or any other unused unit number).

`NEWPAR = 0` : The force constants are read from file *vibpot*, but not updated.

`= 1` : The force constants are read from file *vibpot*, and are updated.

`NRITER` $=-1$: If the observed frequencies are given, the force constant correlation matrix will be printed (by subroutine MOLFIT).

`= 0` : No fit is done.

`=` n (> 0): Number of iterations wanted .

`WEIGHT = 0` : The observed eigenvalues are equally weighted.

`= 1` : The observed eigenvalues are weighted as $1/y$.

`= 2` : The observed eigenvalues are weighted as $1/y^2$.

`MARQ = 1` : The original version of Marquardt optimization method: The initial value of λ is computed once, and is either divided by 3 or multiplied by 2 in subsequent iterations (see sect. 2.5).

`MMARQ = 1` : Modified Marquardt : The value of λ is computed automatically according to a formula before each iteration.

`GAUSS = 1` : Conventional Gauss method (only to be used close to a minimum).

`MGAUSS = 1` : Modified Gauss method (with parameter constraints).

`SAVASS = 0` : No attempt to preserve original order of the eigenvectors.

`= 1` : Program attempts to preserve the order of the eigenvectors (see sect. 2.5).

`PED = 1` : Calculation of potential energy distributions (requires `NRCALC` > 0).

`= 2` : The P.E.D. is expressed only in terms of internal coordinates (i.e., no symmetry coordinates are read).

`FILPED = 1` : The PED w.r.t. force constants will be output in reduced form to a disk file (`pedout`). Only contributions larger than 5 % are taken into account.

`= 2` : The most significant (normalised) PED contributions are written to the disk file `pedout`.

`= 3` : The most significant (unnormalised) PED contributions are written to the disk file `pedout`.

`= 4` : A complete (unnormalised) PED is written to the disk file `pedout`.

`UNITS =` n : Number (m) of molecular asymmetric units (sub-units), used in crystal calculations. If $n < 1$, it will be automatically substituted by 1.

`JACOB` : Indicates how the Jacobian matrix is treated (MOLFIT).

`= 0` : The Jacobian matrix elements will be printed.

`= 1` : The wave-number changes corresponding to a 10 % increase in the force constants will be printed.

`= 2` : The P.E.D. in terms of the force constants will be printed, and also written to a separate file (`pedout`) in a more convenient form.

`= 3` : As `JACOB=2`, but the P.E.D. will also be written to the file `pedout`.

`SCALF` : Used in scaling factor calculations (see sect. 2.8).

`= 1` : The scaling factors will be computed according to Pulay's method (the scaling factors of the non-diagonal force constants are not varied during the calculation).

`= 2` : The scaling factors will be computed according to Pulay's method (and all the scaling factors are varied during the calculation).

`= 3` : Separate scaling factors for diagonal and non-diagonal force constants are to be computed.

`ZMAT = 1` : The atomic coordinates are specified using a Z-matrix (instead of using cartesian coordinates, see sect. 3.3.1).

`INTENS = 1` : The intensities will be calculated, and listed (MOLFIT).

`= 2` : Transition moments w.r.t. normal coordinates are printed.

3.3 Structure of the Data Input to the Subroutines

General :

The sets of data should be input in the following order:

1) Data read by MOLIN.

2) Data read by MOLFIT.

3) Data read by PED.

Numerical parameters are expressed in a free format, (e.g., 1 = 1. = 1.E0) and are separated by at least one space. Non-numeric characters, except +, −, ., E , and space, are assumed to indicate the end of the numerical information.

3.3.1 Input Data for MOLIN

1) Input force constant, or scaling factor definitions (see sect. 3.3.1.1) . End of input is indicated by a blank line or a semicolon on the last line.

Data sets 2) - 6) are repeated if several molecules (or molecular blocks) are considered . If DATFIL is greater than 0 (see sect. 3.2), they are read from a file.

2) Name of the molecule (maximum 80 characters).

3) Dimensions in the format

   N1 N2 N3

   where the numerical parameters N1-N3 have the following meanings:

   N1 = number of atoms in the molecule ($\leq$ 30).

   N2 = 1 : linear molecule.

   N2 = 2 : planar molecule.

   N2 = 3 : a three-dimensional molecule (general case).

   N3 = 1 : denotes isotopically substituted molecule (else this parameter is omitted).

$**$ If the dimension line is blank, control will be transferred to the main program (indicates end of data).

4) Masses of the atoms in free format :

   M1 M2 M3 M4 M5 M5 ...

   The masses may be listed on several consecutive lines (not more than 20 on each).

$**$ If N3 =1 on the dimension line, data sets 5) and 6) will be skipped.

$**$ If `BMAT = 1`, data set 5) is included, otherwise 6a) and 6b) .

5) The B-matrix input directly: Insert one or several lines with a list of the names of the internal (or symmetry coordinates) in the format

   COORD1, COORD2, COORD3 , COORD4 , . . . , COORDN :

where COORD1,COORD2 etc are symbols separated by commas. Only the 8 left-most characters are considered by the program. The set of coordinate names is concluded by a colon (:). The list of coordinate names is followed by the values of the non-zero B-matrix elements. The line format is

I1 J1 B1 I2 J2 B2 I3 J3 B3 ..

where B1 denotes the value of the B-matrix element B(I1,J1) etc. Not more than about 6 matrix elements should be listed on each line. The set of matrix elements is concluded by a blank line.

6a) The B-matrix is generated by the program : 1) `ZMAT=0`: The cartesian coordinates of the atoms are input in the following format:

ATOM1 : X Y Z N1

where ATOM1 denotes the symbolic name of an atom, and X, Y, and Z denote their cartesian coordinates (usually expressed in Ångströms, 0.1 Nm) . Only the first 6 characters of the name of the atom are considered by the program. N1 is an integer parameter,which is usually omitted, except in the treatment of 'dummy atoms' (associated with ligands in complexes), or 'equivalent' atoms. Equivalent (or extra) atoms are used in crystal calculations to define interactions between different asymmetric units. In this case N1 denotes the sub-unit index of the equivalent atom, and N2 is the number of the corresponding atom in the main unit). The sub-unit index of the main unit is 0, the other units being consecutively numbered 1, 2, 3, ... etc. The atomic coordinates are given on subsequent data lines, and the order of the atoms has to be the same as the order of the atomic masses (data set 3)).

If `NBTYP` is greater than 1, the input format is assumed to be :

ATOM1 : X Y Z C N1

where ATOM1, X, Y, Z, and N1 have the same meanings as before, and C denotes the (partial) charge on ATOM1 , expressed in elementary charge units.

2) `ZMAT` > 0: The atomic coordinates are given as a numerical Z-matrix, in the general format

ATOM1 NUM1 R1 NUM2 A2 NUM3 D3

where ATOM1 denotes the symbolic name of an atom (see above), whose position is defined by three previously defined atoms as follows. ATOM1 is bonded to NUM1 with the bond length R1, which is bonded to NUM2, so that the bond angle between ATOM1, NUM1 and NUM2 is A2. Finally NUM3 is bonded to NUM2, so that the dihedral angle between ATOM1, NUM1, NUM2 and NUM3 is D3. It is equivalent with the definition of a Z-matrix in the Gaussian program manual.

6b) Input of internal coordinate information (cf. sect. 3.1.2) This data set is concluded by a blank line or a semicolon as the last character on the last line.

7) In order to calculate the B-matrix for optically active crystal vibrations (cf. sect. 2.6 and 2.7), the following input must be provided:

a) The non-zero elements of the T_p-matrix are input in the format

I1 J1 T1 I2 J2 T2 I3 J3 T3 ...

where T1 = $T_{I1,J1}$, T2 = $T_{I2,J2}$, etc. Only the non-zero matrix elements need to be given, and the element list for each asymmetric unit is finished by a semicolon (;) as the last character on the line. The T_p-matrices are given for each asymmetric unit in succession, except that the T_p-matrix for the main unit is omitted (since it equals the identity matrix).

b) After this the c_p-coefficients for each symmetry block (Γ) are listed successively, separated by blanks. The list of coefficients for each symmetry block is finished by a semicolon (;), and a blank line indicates the end of input of the combination coefficients.

3.3.1.1 Input of Force Constant Definitions

The force constants are defined by specifying a set of interactions between internal (or symmetry) coordinates or atoms (non-bonded interactions). At most 1000 interactions and 100 force constants can be handled by the program (these dimensions can, however, be changed, as explained in the beginning of the program manual).

In scaling factor calculations, scaling factors for force constants are specified in a similar way, by listing the internal coordinates (and interactions, if `SCALF=3`) that define the corresponding force constants.

A special alpha-numeric code is used to specify the interactions. Each input line is divided into string elements of the form:

$< \textit{Interaction} > (\textit{constant}) < \textit{delimiter} >$

where the delimiter may be one of the following characters:

- indicates a pair of interacting coordinates.

, is used to combine interactions having common force constants.

: ends a list of bonded interactions.

/ ends a list of non-bonded interactions.

A string beginning with a '$*$' in the first column is treated as a comment by the program.

Between the interaction name and the delimiter, a numerical constant may be inserted within parentheses, indicating the value of the constant by which the interaction term in the potential energy has to be multiplied. If omitted, its value is assumed to be 1. Upto eight characters in the coordinate names are considered by the program.

When `NBTYP` is greater than 1, the constant associated with stretch and bend parameters is interpreted as the CFF parameters b_0 and θ_0, resp. For torsional parameters it is interpreted as the angle multiplicity ('k'). For the out-of-plane terms, this constant is zero.

For valence force fields, it is possible to define special coordinate interactions (see sect. 2.3) by enclosing a *type number* within parentheses after the minus sign denoting a pair of interacting coordinates.

Example :

```
CH-(21)CCH : 0 0.05
```

A list of the recognized type numbers follows :

10 : Stretch-stretch interaction (bonds with no common atoms but connected via a common bond).

11 : Stretch-stretch interaction (bonds with common end atoms).

20 : Stretch-bend interaction (bond and angle having common end atoms).

21 : Stretch-bend interaction (bond end atom common with angle center atom).

22 : Stretch-bend interaction (bond included in angle).

30 : Bend-bend interaction (center atoms of the angles connected via a common bond).

31 : Bend-bend interaction (angles having two common end atoms).

32 : Bend-bend interaction (center atom of one angle common with one end atom of the other angle, no common bond).

33 : Bend-bend interaction (angles with common center atoms, no common bond).

34 : Bend-bend interaction (angles with common center atoms and a common bond).

Bend-bend interaction (center atom of one angle common with one end atom of the other angle and a common bond):

35 : *Anti-* (*trans-*) interaction.

36 : *Gauche-* (*cis-*) interaction.

37 : Arbitrary positions of the end atoms.

45 : Interaction between a bond angle and an out-of-plane bend having a common bond.

50 : Interaction between two out-of-plane bend coordinates with one bond in common.

65 : Interaction between a bond angle and a torsion angle having one bond in common.

70 : Interaction between an out-of-plane bend and a torsion with a common central bond.

After the end character, initial values for the associated force constants and their variation ranges should be given in a free format, separated by spaces. One force constant is assumed to be associated with a bonded interaction and two or three with a non-bonded one (see below and the description of the parameter `NBTYP`).

The force constant units should be chosen so that the potential energy will be expressed in units of 1 attojoule (1 mdynÅ). If the force constants are related to energies expressed in kcal/mol, as in the CFF program, they should be divided by the conversion factor 143.9412.

Examples:

```
NH : 1. 0
```

defines a diagonal bonded force constant which is assumed to have the fixed value 1 .

```
NCH/ 0.5 1 −1 0
```

denotes a Urey-Bradley term whose f-constant has the initial value 0.5 and the variation range 1. The ratio f'/f is assumed to have the fixed value -1.

If `NBTYP = 2`, the three parameters associated with a non-bonded interaction coordinate are interpreted as the two Lennard-Jones parameters and the inverse of the dielectric constant in the pair potential

$$A(r/2)^{-12} - B(r/2)^{-6} + 2.30713\ Ce_1e_2r^{-1}$$

where r is the non-bonded distance (computed by the program) and e_1 and e_2 are the charges on the interacting atoms (cf. ref.13).

If `NBTYP = 3`, A and B denote the two parameters in the second form of the Lennard-Jones potential :

$$A[(r/B)^{-12} - 2(r/B)^{-6}]$$

In this case, B is assumed to be given in Ångström units and cannot be varied.

Example (`NBTYP = 3`) :

```
X**X / 0.00208 .001 3.5 0 .333333 0
```

If `SCALF` > 0, the force constant definitions are treated as definitions of *scaling factors* associated with the corresponding internal coordinates (which usually are local symmetry coordinates.

The cartesian force constants in the Gaussian output format (lower triangle) will be read from file 16 (`cartfc`), and transformed into MOLVIB units. Then the cartesian force constants are converted into internal coordinates, and the scaling factors are applied.

3.3.1.2 Input of Internal Coordinate Information

The internal coordinates are also specified by a special alphanumeric code. Each input line consists of a sequence of string elements in the following format:

< *Item* >< *delimiter* >

The 'item' is an alphanumeric symbol. Only the six left-most characters of this symbol are considered by the program. This symbol defines the name of the internal coordinate or atom. The 'delimiter', which determines the type of the item can be one of the following:

- is used to connect the atoms involved (the coordinate type is determined from the number of atoms connected).

: indicates a common name for a group of related coordinates.

/ denotes a torsional , out-of-plane bend coordinate, or non-bonded interaction coordinate (for `NBTYP` > `1`)

, ends information on each independent coordinate.

. (or ,) ends information on each group of related coordinates.

; is used to separate group names, if several group coordinate names are to be listed in the same line.

! denotes an 'extra' stretch coordinate used to indicate a bond (sometimes useful in the construction of crystal B-matrices, for instance). Such a coordinate will be omitted from the list of internal coordinates.

After the delimiters '+', ',' , and '.' it is possible to insert the value of a constant by which all the preceding coordinates in the same line are assumed to be multiplied. If the constant is omitted, it is assumed to have the value 1. The coordinate information should be continued on a new line. The maximum number of related coordinates is 200. MOLVIB knows all the standard internal coordinates (cf. ref. 5). Symmetry coordinates are constructed from linar combinations of the internal coordinates, as shown in the examples.

Examples:

```
NH : N-H1, N-H2 , N-H3
```

defines the group name of three N-H stretch coordinates.

```
HCCH/ H1-C1-C2-H3 + H1-C1-C2-H4 + .25

H2-C1-C2-H3 + H2-C1-C2-H4 . .25
```

denotes the mean of four torsional coordinates.

Observe that the names of the atoms occurring in the coordinate definitions should be the same as those previously defined, and that the names of the coordinates or group coordinates should be consistent with the force constant definitions given earlier.

A special type of coordinate is used to define a dummy atom as a linear combination of a set of given atoms (the centroid of a ring, for instance). Example:

```
X1: C1, C2, C3, C4, C5. 0.2
```

This means that the contributions to the dummy atom `X1` in the B–matrix will be divided by 5, and distributed equally among the five carbon atoms (cf. end of sect. 2.4).

The notation used for the internal coordinates is the following:

Stretch coordinate: `XYs : X1-Y2 .`

Bend coordinate: `XYZb: X1-Y2-Z3 .`

```
                'Middle'
```

Out-of-plane bending coordinate: `XYob: X1-Y2-Z3-Z4 .`

```
              Z4
     X1-Y2 <
              Z3
```

Torsion: `XYt : Z1 -X2-Y3- Z4 .`

```
                   Z4
           X2-Y3 /
      Z1 /
```

Observe that the out-of-plane bend coordinates should be oriented in a consistent way !

3.3.2 Input Data for MOLFIT

If `NEWPAR = 0` , data set 1) is skipped.

1) If updating is wanted, the force constant numbers (N), their new values (P) and variation ranges (D) are input in the following order :

   N1 P1 D1 N2 P2 D2 N3 P3 D3 ...

   End of information is indicated by a semicolon (;) or a blank line.

2) The observed normal frequencies of the molecules are input in the following format:

   X1 M1 N1 X2 M1 N2 ...

   Here X1 denotes a frequency (in cm^{-1}), M1 is the molecule (dynamical matrix) number, and N1 is the corresponding eigenvalue number. The eigenvalues are counted according to their order of magnitude, starting with the null frequencies. Before a frequency fit is attempted, a test diagonalization (using `NRCALC` > 0) should be performed to assure a proper assignment of the frequencies. The maximum number of frequencies is 1000. As explained in the introduction, this dimension can be changed.

   Example:

   ```
   267 1 7 273.8 1 8 312 1 9 387 1 10 623.6 1 11...
   ```

End of information is indicated by a semicolon (;) or a blank line.

If `SCALF` > 0, and a blank line was encountered after the cartesian force constants in file 16 (`cartfc`), the rest of the file is supposed to contain the dipole derivatives (three per line). Isotopically substituted compounds are assumed to have the same dipole derivatives. Infrared intensities will then be calculated from the dipole derivatives, as explained in sect. 2.9. If a line starting with the word `POLAR`, is encountered in file 16 after the dipole derivatives, the following lines in the file are supposed to contain polarization derivatives (used to compute Raman activities).

For several (not isotopically substiuted) molecules the data sets have to be repeated in the mentioned order. The last line may be blank.

3.3.3 Input Data for PED

The elements of the symmetry transformation matrices of all the molecules (except for the isotopically substuted ones) should be given on input.

The format of the symmetry coordinate names is as follows:

SCOOR : I1 U1 I2 U2 I3 U3 ...

Here SCOOR denotes a symmetry coordinate name (observe that upto eight characters are considered), I1, I2, and I3 denote internal coordinate numbers and U1, U2, and U3 are the corresponding matrix elements.

If several symmetry coordinates are listed on the same line, the data sets should be separated by commas (,).

Example:

```
ADS1: 2 1 3 -1, BDS1 : 5 1 6 -1,
```

Symmetry coordinates which are identical to internal coordinates need not to be listed. Observe also that the normalization of the symmetry coordinates is automatically done by the program! If the force field calculation has been done in symmetry coordinates, or if the P.E.D. in terms of internal coordinates is preferred, one should use the control parameter value `PED`$= 2$ to suppress some unnecessary output.

N.B. If the force field is of the CFF type (`NBTYP` > 1), the force field will first be transformed from cartesian coordinates into internal coordinates, by calculating the *pseudo-inverse* of the B-matrix (cf. sect. 2.8). By a further transformation of the force field into symmetry coordinates, and by computing the L-matrix in symmetry coordinates, the P.E.D. can be expressed in symmetry coordinates. Using the control parameter `FILPED`, the P.E.D. can be written to a separate disk file (`pedout`), as explained in sect. 2.

If `SCALF` > 0 the force constants will be output (after scaling) in cartesian coordinates.

4 Input examples

(additional examples can be found among the test inputs)

4.1 Test Example 1 : The Ethane Molecule

```
*** First part of standard M O L V I B test:
% GVFF for ethane (C2H6) -- initial calculation
NRC,JACO=1,NMOL=3,PRINT=1;
*
* Valence force field for ethane
*
CH: 4.8 .1
CC: 4.5 .1
CH-(11)CH : 0 .1
CH-CC : 0 0
HCH : 0.5 .1
CCH : 0.7 .1
CH-(22)HCH : 0.1 .05
CC-CCH : 0.3 .05
CH-(22)CCH : 0 .05
HCH-(34)HCH : 0 .05
CCH-(34)CCH : 0 0
HCH-(34)CCH : 0 0
CCH-(35)CCH : 0.1 .05
CCH-(36)CCH : 0 .05
HCCH : 0.05 .01 ;
CH3CH3 --- Struct. CH=1.095,CC=1.531,HCH=107.4 (Kuchitsu)
8 3
12.011 1.008 1.008 1.008 12.011 1.008 1.008 1.008
C1: 0 0 0
H1: 1.019013 0 -.400795
H2: -.509507 .882491 -.400795
H3: -.509507 -.882491 -.400795
C2: 0 0 1.531
H4: -1.019013 0 1.931795
H5: .509507 -.882491 1.931795
H6: .509507 .882491 1.931795
CH:C1-H1,C1-H2,C1-H3,C2-H4,C2-H5,C2-H6.
CC:C1-C2.
HCH:H1-C1-H2,H2-C1-H3,H3-C1-H1,H4-C2-H5,H5-C2-H6,H6-C2-H4.
CCH:C1-C2-H4,C1-C2-H5,C1-C2-H6,C2-C1-H1,C2-C1-H2,C2-C1-H3.
HCCH/ H1-C1-C2-H4+H2-C1-C2-H4+H3-C1-C2-H4+ .1111111
H1-C1-C2-H5+H2-C1-C2-H5+H3-C1-C2-H5+ .1111111
H1-C1-C2-H6+H2-C1-C2-H6+H3-C1-C2-H6. .1111111;
CH3CD3
8 3 1
12.011 1.008 1.008 1.008 12.011 2.0141 2.0141 2.0141
CD3CD3
8 3 1
12.011 2.0141 2.0141 2.0141 12.011 2.0141 2.0141 2.0141
```

```
*** Second part of standard M O L V I B test:
% GVFF for ethane (C2H6) -- frequency fit and PED
NRCA,NMOLE=0,JACO=2,PRINT=0,NRIT=20,WEIGHT,SAV,MGAUSS,PED;
289 1 7 821 1 8 821 1 9 995 1 10 1183 1 11 1183 1 12
1379 1 17 1388 1 18 1469 1 15 1469 1 16 1472 1 13 1472 1 14
2896 1 19 2954 1 20 2969 1 23 2969 1 24 2985 1 21 2985 1 22
253 2 7 678 2 8 678 2 9 904 2 10 1066 2 11 1066 2 12
1115 2 13 1115 2 14 1122 2 15 1387 2 18 1471 2 16 1471 2 17
2090 2 19 2240 2 20 2240 2 21 2912 2 22 2976 2 23 2976 2 24
200 3 7 594 3 8 594 3 9 843 3 10 970 3 11 970 3 12 1041 3 13
1041 3 14 1081 3 15 1081 3 16 1077 3 17 1155 3 18 2083 3 20 2087 3 19
2225 3 23 2225 3 24 2234 3 21 2234 3 22;
ADS1: 2 1 3 -1, BDS1: 5 1 6 -1,
ADD1: 10 1 8 -1, BDD1: 13 1 11 -1,
ARO1: 18 1 19 -1, BRO1: 15 1 16 -1,
ADS2: 1 2 2 -1 3 -1, BDS2: 4 2 5 -1 6 -1,
ADD2: 9 2 8 -1 10 -1, BDD2: 12 2 11 -1 13 -1,
ARO2: 17 2 18 -1 19 -1, BRO2: 14 2 15 -1 16 -1,
ASS: 1 1 2 1 3 1, BSS: 4 1 5 1 6 1,
ASB: 8 1 9 1 10 1 17 -1 18 -1 19 -1,
BSB: 11 1 12 1 13 1 14 -1 15 -1 16 -1;
```

4.2 Test Example 2 : The B-matrix of crystalline polyethylene.

```
% B-MATRIX FOR POLY-ETHYLENE CRYSTAL -- MOLVIB TEST
NMOLE,UNIT=4,PRINT;

* POLY ETHYLENE *
13 3
12.01115 1.007825 1.007825 12.01115 12.01115 1.007825
1.007825 12.01115 1.007825 1.007825 1.007825 1.007825
1.007825
C0: .3337705 .2901423 -.6254364
H0A: 1.3615136 .0389284 -.6254364
H0B: .2280360 1.3428456 -.6254364
C1’: -.3337705 -.2901423 -1.8763091 1 1
C1: -.3337705 -.2901423 .6254364 1 1
H1A: -1.3615136 -.0389284 .6254364 1 2
H1B: -.2280360 -1.3428456 .6254364 1 3
C0’: .3337705 .2901423 1.8763091 0 1
H3B: 3.8085357 1.0901540 -.6254364 2 3
H4A’: 2.2189861 2.4719283 .6254364 3 2
H4B: 3.3524636 -1.0901540 .6254364 3 3
H4A: 2.2189861 -2.3940709 .6254364 3 2
H1B’: -.2280360 3.5231536 .6254364 1 3
CC: C0-C1,C1’-C0!.
CH2S: C0-H0A+C0-H0B. .7071068
CH2A: C0-H0A+ .7071068
C0-H0B. -.7071068
CCC: C1’-C0-C1+ .9128709
H0A-C0-H0B+C1-C0-H0A+C1-C0-H0B+C1’-C0-H0A+C1’-C0-H0B. -.1825742
CH2B: H0A-C0-H0B+ .8944272
C1-C0-H0A+C1-C0-H0B+C1’-C0-H0A+C1’-C0-H0B. -.2236068
CH2W: C1-C0-H0A+C1-C0-H0B+ .5
C1’-C0-H0A+C1’-C0-H0B. -.5
CH2R: C1-C0-H0A+C1’-C0-H0A+ .5
C1-C0-H0B+C1’-C0-H0B. -.5
CH2T: C1-C0-H0A+C1’-C0-H0B+ .5
C1-C0-H0B+C1’-C0-H0A. -.5
CCTR/ H0A-C0-C1-H1A+H0A-C0-C1-H1B+H0A-C0-C1-C0’+ .1111111
H0B-C0-C1-H1A+H0B-C0-C1-H1B+H0B-C0-C1-C0’+ .1111111
C1’-C0-C1-H1A+C1’-C0-C1-H1B+C1’-C0-C1-C0’. .1111111
H**H: H0A-H4A’,H0A-H4A,H0A-H4B,H0B-H4A’,H0A-H3B,H0B-H1B’. ;
1 1 -1. 2 2 -1. 3 3 1. ;
1 1 1. 2 2 -1. 3 3 1. ;
1 1 -1. 2 2 1. 3 3 1. ;
1. 1. 1. 1.
1. 1. -1. -1.
1. -1. 1. -1.
1. -1. -1. 1.
```

4.3 Test Example 3: A CFF calculation for ethane

```
% ETHANE ... 9 torsions ... PEF 303(opt) ... Trial NCA calculation
NBTYP=2,NMOLE,PRINT,NRCALC;
*
* PEF 303 (opt) converted to MOLVIB units . TS 86-03-06
* ------------------------------------------------------
*
CH(1.099) : 4.65468 0
CC(1.5157): 3.91186 0
CCH(1.910633): .649572 0
CCC(1.910633): .989621 0
HCH(1.910633): .519657 0
HCCH(3.): .00098860 0
C**H,H**C/ .151864 0 .015862 0 .333333 0
C**C/ .530237 0 .038632 0 .333333 0
H**H/.043495 0 .00651314 0 .333333 0

ETHANE 00 01
8 3
12.0110 1.0079 1.0079 1.0079 12.0110
1.0079 1.0079 1.0079
C 1: 0.0000000 -0.0000000 0.7644474 0.0
H 2: 0.0596614 1.0312498 1.1456282 0.0
H 3: -0.9229192 -0.4639566 1.1456282 0.0
H 4: 0.8632578 -0.5672932 1.1456282 0.0
C 5: -0.0000000 -0.0000000 -0.7644474 0.0
H 6: -0.8632578 0.5672932 -1.1456282 0.0
H 7: -0.0596614 -1.0312498 -1.1456282 0.0
H 8: 0.9229192 0.4639566 -1.1456282 0.0
CH:C 1-H 2,C 1-H 3,C 1-H 4,C 5-H 6,C 5-H 7,C 5-H 8,
CC:C 1-C 5,
HCH:H 2-C 1-H 3,H 2-C 1-H 4,H 3-C 1-H 4,H 6-C 5-H 7,H 6-C 5-H 8,
H 7-C 5-H 8,
CCH:C 5-C 1-H 2,C 5-C 1-H 3,C 5-C 1-H 4,C 1-C 5-H 6,C 1-C 5-H 7,
C 1-C 5-H 8,
HCCH/H 6-C 5-C 1-H 2,H 6-C 5-C 1-H 3,H 6-C 5-C 1-H 4,H 7-C 5-C 1-H 2,
H 7-C 5-C 1-H 3,H 7-C 5-C 1-H 4,H 8-C 5-C 1-H 2,H 8-C 5-C 1-H 3,
H 8-C 5-C 1-H 4,
H**H/H 2-H 6,H 2-H 7,H 2-H 8,H 3-H 6,H 3-H 7,H 3-H 8,H 4-H 6,H 4-H 7,
H 4-H 8,
```

4.4 Test Example 4 : The Formaldehyde Molecule
(scaling factor calculation)

```
% Scaling factor test for FORMALDEHYDE * TS 1997-5-9, 1999-10-23
NRCA,JACO=1,NMOLE=2,PRINT,SCALF=2,ZMAT,INTEN,NRIT=10,MGAUS,WEIGHT,SAVA;
S1,S4 : 1 0.05
S2 : 1. 0.05
S3,S5: 1. 0.05
S6 : 1. 0.05;
Formaldehyde, becke3lyp/6-31g**, frequencies and dip.derivs.
4 3
15.99491 12 1.00783 1.00783
O1
C2 1 1.20678664
H3 2 1.11032398 1 122.37705146
H4 2 1.11032398 1 122.37705146 3 180.
S1: C2-H3 + C2-H4. .707107
S2: C2-O1.
S3: H3-C2-H4 + .816497
H3-C2-O1 + H4-C2-O1. -.408248
S4: C2-H3 + .707107
C2-H4. -.707107
S5: H3-C2-O1 + .707107
H4-C2-O1. -.707107
S6: O1-C2-H4-H3. ;
D2CO
4 3 1
15.99491 12 2.0141 2.0141
1191 1 7 1287.7 1 8 1562.6 1 9 1763.7 1 10 2944.3 1 11 3008.7 1 12
955 2 7 1014.3 2 8 1139.6 2 9 1716.7 2 10 2143.5 2 11 2254.5 2 12 ;
* "harmonized" freq. from Duncan & Mallinson
```

Ab initio force constants and dipole derivatives for formaldehyde

```
0.07368201
0. 0.01901460
0. 0. 0.89884867
-0.10776412 0. 0. 0.58164862
0. -0.05758915 0. 0. 0.17426055
0. 0. -0.81484215 0. 0.
1.01577283
0.01704105 0. -0.01837269 -0.23694225 0.
-0.07591285 0.23894077
0. 0.01928728 0. 0. -0.05833570
0. 0. 0.01949797
-0.03660268 0. -0.04200326 -0.08095982 0.
-0.10046534 0.10592402 0. 0.13205032
0.01704105 0. 0.01837269 -0.23694225 0.
0.07591285 -0.01903957 0. 0.01163849 0.23894077
0. 0.01928728 0. 0. -0.05833570
0. 0. 0.01955046 0. 0.
0.01949797
```

```
0.03660268 0. -0.04200326 0.08095982 0.
-0.10046534 -0.01163849 0. 0.01041828 -0.10592402
0. 0.13205032

-0.4247792 0. 0.
0. -0.3116738 0.
0. 0. -0.8608323
0.8605767 0. 0.
0. 0.1796489 0.
0. 0. 0.9888199
-0.2178988 0. -0.1143274
0. 0.0660125 0.
-0.1071656 0. -0.0639938
-0.2178988 0. 0.1143274
0. 0.0660125 0.
0.1071656 0. -0.0639938
```

4.5 Test Example 5: Li-cyclopentadienyl complex
(scaling factors and dummy atom)

```
% Scaling factors for Li-cyclopentadienyl (test)
NRCALC,NMOLE=1,PRINT=1,JACOB=1,SCALF;
LiX: 1. 0.
CR12,CR23,CR34,CR45,CR51: 1. 0.
CRH1,CRH2,CRH3,CRH4,CRH5: 1. 0.
CA1,CA2,CA3,CA4,CA5: 1. 0.
CBH1,CBH2,CBH3,CBH4,CBH5: 1. 0.
LiXC: 1. 0.
CGH1,CGH2,CGH3,CGH4,CGH5: 1. 0.
CT12,CT23,CT34,CT45,CT51: 1. 0.;
* C5H5-Li, geometry from HF/6-31G*
12 3
7.016 12 12 12 12 12
1.00783 1.00783 1.00783 1.00783 1.00783 1
Li : 0.000000 0.000000 0.000000
C1 : 1.199438 0.000000 1.762044
C2 : 0.370666 1.140718 1.762044
C3 : -0.970326 0.705011 1.762044
C4 : -0.970333 -0.704990 1.762044
C5 : 0.370655 -1.140711 1.762044
H6 : 2.273130 -0.000004 1.762044
H7 : 0.702459 2.161859 1.762044
H8 : -1.838959 1.336115 1.762044
H9 : -1.838971 -1.336086 1.762044
H10 : 0.702440 -2.161853 1.762044
X : 0.000000 0.000000 1.762044 1
LiX: Li-X.
XC: X-C1!,X-C2!,X-C3!,X-C4!,X-C5!.
X: C1,C2,C3,C4,C5. 0.2
CRH1: C1-H6.
CRH2: C2-H7.
CRH3: C3-H8.
CRH4: C4-H9.
CRH5: C5-H10.
CR12: C1-C2.
CR23: C2-C3.
CR34: C3-C4.
CR45: C4-C5.
CR51: C5-C1.
CA1: C5-C1-C2,
CA2: C1-C2-C3,
CA3: C2-C3-C4.
CA4: C3-C4-C5.
CA5: C4-C5-C1.
CBH1: H6-C1-C2+ 0.5
H6-C1-C5. -0.5
CBH2: H7-C2-C3+ 0.5
H7-C2-C1. -0.5
CBH3: H8-C3-C4+ 0.5
```

```
H8-C3-C2. -0.5
CBH4: H9-C4-C5+ 0.5
H9-C4-C3. -0.5
CBH5: H10-C5-C1+ 0.5
H10-C5-C4. -0.5
LiXC: Li-X-C1, Li-X-C2,
Li-X-C3, Li-X-C4, Li-X-C5,
CGH1: H6-C1-C2-C5.
CGH2: H7-C2-C3-C1.
CGH3: H8-C3-C4-C2.
CGH4: H9-C4-C5-C3.
CGH5: H10-C5-C1-C4.
CT12/ H7-C2-C1-H6+ 0.5
C3-C2-C1-C5. 0.5
CT23/ H8-C3-C2-H7+ 0.5
C4-C3-C2-C1. 0.5
CT34/ H9-C4-C3-H8+ 0.5
C5-C4-C3-C2. 0.5
CT45/ H10-C5-C4-H9+ 0.5
C1-C5-C4-C3. 0.5
CT51/ H6-C1-C5-H10+ 0.5
C2-C1-C5-C4. 0.5;
```