

Vibrational Analysis in *Gaussian*

Joseph W. Ochterski, Ph.D.
help@gaussian.com

October 29, 1999
Minor corrections 14 June 2018

Abstract

One of the most commonly asked questions about *Gaussian* is “What is the definition of reduced mass that *Gaussian* uses, and why is it different than what I calculate for diatomics by hand?” The purpose of this document is to describe how *Gaussian* calculates the reduced mass, frequencies, force constants, and normal coordinates which are printed out at the end of a frequency calculation.

Contents

1	The short answer	1
2	The long answer	2
2.1	Mass weight the Hessian and diagonalize	2
2.2	Determine the principal axes of inertia	3
2.3	Generate coordinates in the rotating and translating frame	4
2.4	Transform the Hessian to internal coordinates and diagonalize	5
2.5	Calculate the frequencies	5
2.6	Calculate reduced mass, force constants and cartesian displacements	5
3	Summary	7
4	A note about low frequencies	8
5	Acknowledgements	10

1 The short answer

So why is the reduced mass different in *Gaussian*? The short answer is that *Gaussian* uses a coordinate system where the normalized cartesian displacement is one unit. This differs from the coordinate system used in most texts, where a unit step of one is used for the change in interatomic distance (in a diatomic). The vibrational analysis of polyatomics in *Gaussian* is not different from that described in “Molecular Vibrations” by Wilson, Decius

and Cross. Diatomics are simply treated the same way as polyatomics, rather than using a different coordinate system.

2 The long answer

In this section, I'll describe exactly how frequencies, force constants, normal modes and reduced mass are calculated in *Gaussian*, starting with the Hessian, or second derivative matrix. I'll outline the general polyatomic case, leaving out details for dealing with frozen atoms, hindered rotors and the like.

I will try to stick close to the notation used in "Molecular Vibrations" by Wilson, Decius and Cross. I will add some subscripts to indicate which coordinate system the matrix is in.

There is an important point worth mentioning before starting. Vibrational analysis, as it's described in most texts and implemented in *Gaussian*, is valid only when the first derivatives of the energy with respect to displacement of the atoms are zero. In other words, the geometry used for vibrational analysis *must* be optimized at the same level of theory and with the same basis set that the second derivatives were generated with. Analysis at transition states and higher order saddle points is also valid. Other geometries are not valid. (There are certain exceptions, such as analysis along an IRC, where the non-zero derivative can be projected out.) For example, calculating frequencies using HF/6-31g* on MP2/6-31G* geometries is not well defined.

Another point that is sometimes overlooked is that frequency calculations need to be performed with a method suitable for describing the particular molecule being studied. For example, a single reference method, such as Hartree-Fock (HF) theory is not capable of describing a molecule that needs a multireference method. One case that comes to mind is molecules which are in a $^2\Pi$ ground state. Using a single reference method will yield different frequencies for the Π_x and Π_y vibrations, while a multireference method shows the cylindrical symmetry you might expect. This is seldom a large problem, since the frequencies of the other modes, like the stretching mode, are still useful.

2.1 Mass weight the Hessian and diagonalize

We start with the Hessian matrix \mathbf{f}_{CART} , which holds the second partial derivatives of the potential V with respect to displacement of the atoms in cartesian coordinates (CART):

$$f_{\text{CART}ij} = \left(\frac{\partial^2 V}{\partial \xi_i \partial \xi_j} \right)_0 \quad (1)$$

This is a $3N \times 3N$ matrix (N is the number of atoms), where $\xi_1, \xi_2, \xi_3 \dots \xi_{3N}$ are used for the displacements in cartesian coordinates, $\Delta x_1, \Delta y_1, \Delta z_1, \dots \Delta z_N$. The $()_0$ refers to the fact that the derivatives are taken at the equilibrium positions of the atoms, and that the first derivatives are zero.

The first thing that *Gaussian* does with these force constants is to convert them to mass

weighted cartesian coordinates (MWC).

$$f_{\text{MWC}ij} = \frac{f_{\text{CART}ij}}{\sqrt{m_i m_j}} = \left(\frac{\partial^2 V}{\partial q_i \partial q_j} \right)_0 \quad (2)$$

where $q_1 = \sqrt{m_1} \xi_1 = \sqrt{m_1} \Delta x_1$, $q_2 = \sqrt{m_1} \xi_2 = \sqrt{m_1} \Delta y_1$ and so on, are the mass weighted cartesian coordinates.

A copy of \mathbf{f}_{MWC} is diagonalized, yielding a set of $3N$ eigenvectors and $3N$ eigenvalues. The eigenvectors, which are the normal modes, are discarded; they will be calculated again after the rotation and translation modes are separated out. The roots of the eigenvalues are the fundamental frequencies of the molecule. *Gaussian* converts them to cm^{-1} , then prints out the $3N$ (up to 9) lowest. The output for water HF/3-21G* looks like this:

```
Full mass-weighted force constant matrix:
Low frequencies --- -0.0008  0.0003  0.0013  40.6275  59.3808  66.4408
Low frequencies --- 1799.1892 3809.4604 3943.3536
```

In general, the frequencies for rotation and translation modes should be close to zero. If you have optimized to a transition state, or to a higher order saddle point, then there will be some negative frequencies which may be listed before the “zero frequency” modes. (Frequencies which are printed out as negative are really imaginary; the minus sign is simply a flag to indicate that this is an imaginary frequency.) There is a discussion about how close to zero is close enough, and what to do if you are not close enough in Section 4 of this paper.

You should compare the lowest real frequencies list in this part of the output with the corresponding frequencies later in the output. The later frequencies are calculated after projecting out the translational and rotational modes. If the corresponding frequencies in both places are not the same, then this is an indication that these modes are contaminated by the rotational and translational modes.

2.2 Determine the principal axes of inertia

The next step is to translate the center of mass to the origin, and determine the moments and products of inertia, with the goal of finding the matrix that diagonalizes the moment of inertia tensor. Using this matrix we can find the vectors corresponding to the rotations and translations. Once these vectors are known, we know that the rest of the normal modes are vibrations, so we can distinguish low frequency vibrational modes from rotational and translational modes.

The center of mass (\mathbf{R}_{COM}) is found in the usual way:

$$\mathbf{R}_{\text{COM}} = \frac{\sum_{\alpha} m_{\alpha} \mathbf{r}_{\alpha}}{\sum_{\alpha} m_{\alpha}} \quad (3)$$

where the sums are over the atoms, α . The origin is then shifted to the center of mass $\mathbf{r}_{\text{COM}\alpha} = \mathbf{r}_{\alpha} - \mathbf{R}_{\text{COM}}$. Next we have to calculate the moments of inertia (the diagonal elements) and the products of inertia (off diagonal elements) of the moment of inertia tensor (\mathbf{I}).

$$\mathbf{I} = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix} = \begin{pmatrix} \sum_{\alpha} m_{\alpha} (y_{\alpha}^2 + z_{\alpha}^2) & -\sum_{\alpha} m_{\alpha} (x_{\alpha} y_{\alpha}) & -\sum_{\alpha} m_{\alpha} (x_{\alpha} z_{\alpha}) \\ -\sum_{\alpha} m_{\alpha} (y_{\alpha} x_{\alpha}) & \sum_{\alpha} m_{\alpha} (x_{\alpha}^2 + z_{\alpha}^2) & -\sum_{\alpha} m_{\alpha} (y_{\alpha} z_{\alpha}) \\ -\sum_{\alpha} m_{\alpha} (z_{\alpha} x_{\alpha}) & -\sum_{\alpha} m_{\alpha} (z_{\alpha} y_{\alpha}) & \sum_{\alpha} m_{\alpha} (x_{\alpha}^2 + y_{\alpha}^2) \end{pmatrix} \quad (4)$$

This symmetric matrix is diagonalized, yielding the principal moments (the eigenvalues \mathbf{I}') and a 3×3 matrix (\mathbf{X}), which is made up of the normalized eigenvectors of \mathbf{I} . The eigenvectors of the moment of inertia tensor are used to generate the vectors corresponding to translation and infinitesimal rotation of the molecule in the next step.

2.3 Generate coordinates in the rotating and translating frame

The main goal in this section is to generate the transformation \mathbf{D} from mass weighted cartesian coordinates to a set of $3N$ coordinates where rotation and translation of the molecule are separated out, leaving $3N - 6$ or $3N - 5$ modes for vibrational analysis. The rest of this section describes how the Sayvetz conditions are used to generate the translation and rotation vectors.

The three vectors (\mathbf{D}_1 , \mathbf{D}_2 , \mathbf{D}_3) of length $3N$ corresponding to translation are trivial to generate in cartesian coordinates. They are just $\sqrt{m_i}$ times the corresponding coordinate axis. For example, for water (using $m_{\text{H}} = 1$ and $m_{\text{O}} = 16$) the translational vectors are:

$$\begin{aligned}\mathbf{D}_1 &= (1, 0, 0, 4, 0, 0, 1, 0, 0)^t \\ \mathbf{D}_2 &= (0, 1, 0, 0, 4, 0, 0, 1, 0)^t \\ \mathbf{D}_3 &= (0, 0, 1, 0, 0, 4, 0, 0, 1)^t\end{aligned}$$

Generating vectors corresponding to rotational motion of the atoms in cartesian coordinates is a bit more complicated. The vectors for these are defined this way:

$$\begin{aligned}D_{4j,i} &= ((P_y)_i X_{j,3} - (P_z)_i X_{j,2}) / \sqrt{m_i} \\ D_{5j,i} &= ((P_z)_i X_{j,1} - (P_x)_i X_{j,3}) / \sqrt{m_i} \\ D_{6j,i} &= ((P_x)_i X_{j,2} - (P_y)_i X_{j,1}) / \sqrt{m_i}\end{aligned}\tag{5}$$

where $j = x, y, z$; i is over all atoms and P is the dot product of \mathbf{R} (the coordinates of the atoms with respect to the center of mass) and the corresponding row of \mathbf{X} , the matrix used to diagonalize the moment of inertia tensor \mathbf{I} .

The next step is to normalize these vectors. If the molecule is linear (or is a single atoms), any vectors which do not correspond to translational or rotational normal modes are removed. The scalar product is taken of each vector with itself. If it is zero (or very close to it), then that vector is not an actual normal mode and it is eliminated. (If the scalar product is zero, this mode will disappear when the transformation from mass weighted to internal coordinates is done, in Equation 6.) Otherwise, the vector is normalized using the reciprocal square root of the scalar product. *Gaussian* then checks to see that the number of rotational and translational modes is what's expected for the molecule, three for atoms, five for linear molecules and six for all others. If this is not the case, *Gaussian* prints an error message and aborts.

A Schmidt orthogonalization is used to generate $N_{vib} = 3N - 6$ (or $3N - 5$) remaining vectors, which are orthogonal to the five or six rotational and translational vectors. The result is a transformation matrix \mathbf{D} which transforms from mass weighted cartesian coordinates \mathbf{q} to internal coordinates $\mathbf{S} = \mathbf{D}\mathbf{q}$, where rotation and translation have been separated out.

2.4 Transform the Hessian to internal coordinates and diagonalize

Now that we have coordinates in the rotating and translating frame, we need to transform the Hessian, \mathbf{f}_{MWC} (still in mass weighted cartesian coordinates), to these new internal coordinates (INT). Only the N_{vib} coordinates corresponding to internal coordinates will be diagonalized, although the full $3N$ coordinates are used to transform the Hessian.

The transformation is straightforward:

$$\mathbf{f}_{\text{INT}} = \mathbf{D}^\dagger \mathbf{f}_{\text{MWC}} \mathbf{D} \quad (6)$$

The $N_{\text{vib}} \times N_{\text{vib}}$ submatrix of \mathbf{f}_{INT} , which represents the force constants internal coordinates, is diagonalized yielding N_{vib} eigenvalues $\lambda = 4\pi^2\nu^2$, and N_{vib} eigenvectors. If we call the transformation matrix composed of the eigenvectors \mathbf{L} , then we have

$$\mathbf{L}^\dagger \mathbf{f}_{\text{INT}} \mathbf{L} = \mathbf{\Lambda} \quad (7)$$

where $\mathbf{\Lambda}$ is the diagonal matrix with eigenvalues λ_i .

2.5 Calculate the frequencies

At this point, the eigenvalues need to be converted frequencies in units of reciprocal centimeters. First we change from frequencies (ν_i) to wavenumbers ($\tilde{\nu}_i$), via the relationship $\nu_i = \tilde{\nu}_i c$, where c is the speed of light. Solving $\lambda = 4\pi^2\tilde{\nu}^2 c^2$ for $\tilde{\nu}_i^2$ we get

$$\tilde{\nu}_i = \sqrt{\frac{\lambda_i}{4\pi^2 c^2}} \quad (8)$$

The rest is simply applying the appropriate conversion factors: from a single molecule to a mole, from hartrees to joules, and from atomic mass units to kilograms. For negative eigenvalues, we calculate $\tilde{\nu}_i$ using the absolute value of λ_i , then multiply by -1 to make the frequency negative (which flags it as imaginary). After this conversion, the frequencies are ready to be printed out.

2.6 Calculate reduced mass, force constants and cartesian displacements

All the pieces are now in place to calculate the reduced mass, force constants and cartesian displacements. Combining Equation 6 and Equation 7, we arrive at

$$\mathbf{L}^\dagger \mathbf{D}^\dagger \mathbf{f}_{\text{MWC}} \mathbf{D} \mathbf{L} = \mathbf{\Lambda} = \mathbf{l}_{\text{MWC}}^\dagger \mathbf{f}_{\text{MWC}} \mathbf{l}_{\text{MWC}} \quad (9)$$

where $\mathbf{l} = \mathbf{D} \mathbf{L}$ is the matrix needed to diagonalize \mathbf{f}_{MWC} . Actually, \mathbf{l}_{MWC} is never calculated directly in *Gaussian*. Instead, $\mathbf{l}_{\text{CART}} = \mathbf{M} \mathbf{D} \mathbf{L}$ is calculated, where \mathbf{M} is a diagonal matrix defined by:

$$M_{i,i} = 1/\sqrt{m_i} \quad (10)$$

and i runs over the x , y , and z coordinates for every atom. The individual elements of \mathbf{l}_{CART} are given by:

$$l_{\text{CART}k,i} = \sum_j^{3N} \left(\frac{D_{k,j} L_{j,i}}{\sqrt{m_k}} \right). \quad (11)$$

The column vectors of these elements, which are the normal modes in cartesian coordinates, are used in several ways. First of all, once normalized by the procedure described below, they are the displacements in cartesian coordinates. Secondly, they are useful for calculating a number of spectroscopic properties, including IR intensities, Raman activities, depolarizations and dipole and rotational strengths for VCD.

Normalization is a relatively straight forward process. Before it is printed out, each of the $3N$ elements of $\mathbf{l}_{\text{CART}i}$ is scaled by normalization factor \mathcal{N}_i , for that particular vibrational mode. The normalization is defined by:

$$\mathcal{N}_i = \sqrt{\left(\sum_k^{3N} l_{\text{CART}k,i}^2 \right)^{-1}} \quad (12)$$

The reduced mass μ_i for the vibrational mode is calculated in a similar fashion:

$$\mu_i = \left(\sum_k^{3N} l_{\text{CART}k,i}^2 \right)^{-1} = \left(\sum_k^{3N} \left(\frac{l_{\text{MWC}k,i}}{\sqrt{m_k}} \right)^2 \right)^{-1} = \left(\sum_k^{3N} \left(\frac{l_{\text{MWC}k,i}^2}{m_k} \right) \right)^{-1} = \mathcal{N}_i^2 \quad (13)$$

Note that since \mathbf{D} is orthonormal, and we can (and do) choose \mathbf{L} to be orthonormal, then \mathbf{I} is orthonormal as well. (Since $\mathbf{D}^\dagger \mathbf{D} = \mathbf{1}$, $\mathbf{L}^\dagger \mathbf{L} = \mathbf{1}$ then $\mathbf{I}^\dagger \mathbf{I} = (\mathbf{D}\mathbf{L})^\dagger \mathbf{D}\mathbf{L} = \mathbf{L}^\dagger \mathbf{D}^\dagger \mathbf{D}\mathbf{L} = \mathbf{L}^\dagger \mathbf{1}\mathbf{L} = \mathbf{1}$).

We now have enough information to explain the difference between the reduced mass *Gaussian* prints out, and the one calculated using the formula usually used for diatomics:

$$\frac{1}{\mu} = \frac{1}{m_1} + \frac{1}{m_2} \quad (14)$$

The difference is in the numerator of each term in the summation. *Gaussian* uses $l_{\text{MWC}k,i}^2$ rather than 1. Using the elements of \mathbf{l}_{MWC} yields the consistent results for polyatomic cases, and automatically takes symmetry into consideration. Simply extending the formula from Equation 14 to $\frac{1}{\mu} = \sum_i^{\text{atoms}} \frac{1}{m_i}$ would (incorrectly) yield the same reduced mass for every mode of a polyatomic molecule.

The effect of using the elements of \mathbf{l}_{MWC} in the numerator is to make the unit length of the coordinate system *Gaussian* uses be the normalized cartesian displacement. In other words, in the coordinate system that *Gaussian* uses, the sum of the squares of the cartesian displacements is 1. (You can check this in the output). In the more common coordinate system for diatomics, the unit length is a unit change in internuclear distance from the equilibrium value.

One of the consequences of using this coordinate system is that force constants which you think should be equal are not. A simple example is H_2 versus HD . Since the Hessian depends only on the electronic part of the Hamiltonian, you would expect the force constants to be

the same for these to molecules. In fact, the force constant *Gaussian* prints out is different. The different masses of the atoms leads to a different set of Sayvetz conditions, which in turn, change the internal coordinate system the force constants are transformed to, and ultimately the resulting force constant.

The coordinates used to calculate the force constants, the reduced mass and the cartesian displacements are all internally consistent. The force constants k_i are given by $k_i = 4\pi^2\tilde{\nu}^2\mu_i$, since $\tilde{\nu} = \frac{1}{2\pi}\sqrt{\frac{k_i}{\mu_i}}$. The force constants are converted from atomic units to millidyne/angstrom.

3 Summary

To summarize, the steps *Gaussian* uses to perform vibrational analysis are:

1. Mass weight the Hessian

$$f_{\text{MWC}ij} = \frac{f_{\text{CART}ij}}{\sqrt{m_i m_j}}$$

2. Determine the principal axes of inertia

$$\mathbf{I}' = \mathbf{X}^\dagger \mathbf{I} \mathbf{X}$$

3. Generate coordinates in the rotating and translating frame

$$\mathbf{S} = \mathbf{D} \mathbf{q}$$

4. Transform the Hessian to internal coordinates and diagonalize

$$\mathbf{f}_{\text{INT}} = \mathbf{D}^\dagger \mathbf{f}_{\text{MWC}} \mathbf{D}$$

$$\mathbf{L}^\dagger \mathbf{f}_{\text{INT}} \mathbf{L} = \mathbf{\Lambda}$$

5. Calculate the frequencies

$$\tilde{\nu}_i = \sqrt{\frac{\lambda_i}{4\pi^2 c^2}}$$

6. Calculate reduced mass, force constants and cartesian displacements

$$\mu_i = \left(\sum_k^{3N} l_{\text{CART}k,i}^2 \right)^{-1}$$

$$k_i = 4\pi^2 \tilde{\nu}_i^2 c^2 \mu_i$$

$$\mathbf{l}_{\text{CART}} = \mathbf{M} \mathbf{D} \mathbf{L}$$

Criteria	Low frequencies					
Opt	-0.0008	0.0003	0.0013	40.6275	59.3808	66.4408
Opt=Tight	0.0011	0.0013	0.0015	4.1908	-6.8779	12.4224
Opt=VeryTight	-0.0011	0.0014	0.0015	-0.9207	-1.1831	-1.6023

Table 1: The effect of optimization criteria on the low frequencies of water using HF/3-21G*. The frequencies are sorted by increasing absolute value, so that it’s easier to distinguish rotational modes from vibrational modes.

4 A note about low frequencies

You’ll find that the frequencies for the translations are almost always extremely close to zero. The frequencies for rotations are quite a bit larger. So, how “close to zero” is close enough? For most methods (HF, MP2, etc.), you’d like the rotational frequencies to be around 10 wavenumbers or less. For methods which use numerical integration, like DFT, the frequencies should be less than a few tens of wavenumbers, say 50 or so.

If the frequencies for rotations are not close to zero, it may be a signal that you need to do a tighter optimization. There are a couple of ways to accomplish this. For most methods, you can use `Opt=Tight` or `Opt=Verytight` on the route card to specify that you’d like to use tighter convergence criteria. For DFT, you may also need to specify `Int=Ultrafine`, which uses a more accurate numerical integration grid.

As an example, I reran the water HF/3-21G* calculation above, with both `Opt=Tight` and `Opt=VeryTight`. You can see in Table 1 that the rotational frequencies are an order of magnitude better for `Opt=Tight` than they were for just `Opt`. Using `Opt=Verytight` makes them even better. This raises the question of whether the you need to use tighter convergence. The answer is: it depends – different users will be interested in different results. There is a trade off between accuracy and speed. Using `Opt=Tight` or `Int=Ultrafine` makes the calculation take longer in addition to making the results more accurate. The default convergence criteria are set to give an accuracy good enough for most purposes without spending time to converge the results beyond this accuracy. You may find that you need to use the tighter criteria to compare to spectroscopic values, or to resolve a structure with a particularly flat potential energy surface.

In the water frequency calculation above, using tighter convergence criteria makes almost no difference in terms of energy or bond lengths, as Table 2 demonstrates. The energy is converged to less than 1 microHartree, and the OH bond length is converged to 0.0002 angstroms. Tightening up the convergence criteria is useful for getting a couple of extra digits of precision in the symmetric stretch frequency.

You can also see that the final geometry parameters obtained with the default optimization criteria depend somewhat on the initial starting geometry. Using `Opt=VeryTight` all but eliminates these differences. I’ve included the starting geometries in Table 3, for those who wish to reproduce these results. (Using the default convergence criteria may give somewhat different results than those I’ve shown if you use a different machine, or even the same machine using different libraries or a different version of the compiler).

With DFT, `Opt=VeryTight` alone is not necessarily enough to converge the geometry to

Criteria	Start Geom	Energy	R_{OH}	A_{HOH}	Bend	Symmetric Stretch	Antisymm. Stretch
Opt	A	-75.5859596012	0.9669	107.7241	1799.1892	3809.4604	3943.3536
Opt	B	-75.5859596488	0.9665	107.6348	1799.5957	3814.2216	3947.2011
Opt=Tight	A	-75.5859597578	0.9667	107.6784	1799.3335	3812.3499	3945.7723
Opt=Tight	B	-75.5859597580	0.9667	107.6811	1799.3156	3812.2440	3945.6938
Opt=Verytight	A	-75.5859597582	0.9667	107.6818	1799.2877	3812.3779	3945.8339
Opt=Verytight	B	-75.5859597582	0.9667	107.6820	1799.2854	3812.3847	3945.8418

Table 2: The default optimization settings yield results accurate enough for most purposes. Tighter optimizations make almost no difference for this HF/3-21G* frequency calculation on water.

Geometry A	Geometry B
O	O
H,1,R2	H,1,R2
H,1,R2,2,A3	H,1,R2,2,A3
R2=0.96	R2=1.0
A3=109.47122063	A3=109.5

Table 3: Initial geometries for water optimization calculations. Geometry A was produced by `Geom=ModelA`. Geometry B is a slightly modified version of Geometry A.

the point where the low frequencies are as close to zero as you would like. To demonstrate this, I have run B3LYP/3-21G* optimizations on water, starting with geometry B from Table 3, with `Opt`, `Opt=Tight` and `Opt=VeryTight`. The results are in Table 4. The low frequencies from these two jobs hardly change, and in fact get worse for the `Tight` and `VeryTight` optimizations.

Given the straight forward convergence seen with Hartree-Fock theory, this might not seem to make sense. However, it does make sense if you recall that DFT is done using a numerical integration on a grid of points. The accuracy of the default grid is not high enough for computing low frequency modes very precisely. The solution is to use a more numerically accurate grid. The tighter the optimization criteria, the more accurate the grid needs to be. As you can see in Table 4, increasing the convergence criteria from `Tight` to `VeryTight` without increasing the numerical accuracy of the grid yields no improvement in the low frequencies. For `Opt=Tight`, we recommend using the `Ultrafine` grid. This is a good combination to use for systems with hindered rotors, or if exact conformation is of concern. If still more accuracy is necessary, then an unpruned `199974` grid can be used with `Opt=VeryTight`. Again, the higher accuracy comes at a higher cost in terms of CPU time. The `VeryTight` optimization with a `199974` grid is *very* expensive, even for medium sized molecules. The default grids are accurate enough for most purposes.

Criteria	Grid	Low frequencies					
Opt	FineGrid	-0.0005	-0.0011	0.0011	16.5776	17.8265	-38.2354
Opt=Tight	FineGrid	-0.0011	-0.0011	0.0005	25.8979	-29.0202	37.0170
Opt=VeryTight	FineGrid	-0.0006	-0.0007	-0.0008	25.8977	-29.0203	37.0168
Opt=Tight	UltraFine	0.0012	0.0022	0.0024	-1.5386	-4.8182	-9.0313
Opt=VeryTight	UltraFine	0.0012	0.0022	0.0024	-1.5386	-4.8182	-9.0313
Opt=VeryTight	Grid=199974	-0.0004	0.0010	0.0011	-4.8721	-5.3561	-6.3672

Table 4: The effect of grid size on the low frequencies from B3LYP/3-21G* on water with Opt, Opt=Tight and Opt=VeryTight. More accurate grids are necessary for a truly converged optimization. The frequencies are sorted by increasing absolute value, so that it's easier to distinguish rotational modes from vibrational modes.

5 Acknowledgements

I'd like to thank John Montgomery for his constructive suggestions, Michael Frisch for clarifying several points, and H. Berny Schlegel for taking the time to discuss this material with me. Thanks also to Jim Cheeseman, for lending me his copy of Wilson, Decius and Cross.