
PVSCF V0.56 Manual
Parallel Vibrational Self-Consistent Field

David M. Benoit* and Christopher G. P. Jay
Department of Chemistry, University of Hull, UK

Inga Respondek
*Nachwuchsgruppe Theorie – SFB 569, University of Ulm,
Albert-Einstein-Allee 11, D-89081 Ulm, Germany*

Bruno O. Madebène
*L.A.D.I.R Boite 49, Bat F74 3eme étage, Université Pierre et Marie Curie,
4 place Jussieu, F-75252 Paris, France*

Yohann S. Scribano
*Laboratoire J.A.Dieudonné, UMR CNRS 6621 Université de Nice Sophia-Antipolis,
Parc Valrose, F-06108 Nice Cedex 2, France*

David M. Lauvergnat
*Laboratoire de Chimie Physique, UMR 8000 Université Paris-Sud-11,
F-91405 Orsay Cedex, France*

Thursday 5th March, 2015

*Electronic address: d.benoit@hull.ac.uk

Contents

1	How PVSCF works	3
1.1	Methodology of the VSCF approach	3
1.2	Implementation of the calculation of the total energy	3
1.3	Vibrational excited states from perturbation theory	4
1.4	Vibrational excited states from the interaction of configurations	5
1.5	Calculation of harmonic frequency and anharmonicity constants	7
1.6	Construction of the Hessian matrix and projection of translations and rotations	7
2	Getting started with and using PVSCF	10
2.1	Introduction to PVSCF	10
2.2	Generating a potential energy surface	17
2.3	Diagonal frequencies	18
2.4	Template files	19
2.5	Hessian calculation	21
2.6	Post-VSCF calculations	24
2.7	TDVCI	25
2.8	FAST-VSCF methods	25
2.8.1	What is implemented?	26
2.9	Saving wavefunctions	26
2.10	Optimisation of PES sampling	26
2.11	Degenerate perturbation theory	27
2.12	Interfacing with <i>ab initio</i> programs	28
2.12.1	Orca	28
2.12.2	GAMESS-US	28
2.12.3	NWChem	29
2.12.4	CPMD	29
2.12.5	CP2K	29

	2
2.12.6 PSI3	31
2.12.7 DALTON2	31
2.12.8 SIESTA	32
2.13 Curvilinear coordinate VSCF	32
2.13.1 Curvilinear coordinates: a practical guide	32
3 PVSCF usage	33
3.1 VCI calculations	33
3.2 VMP2	33
3.3 Constants	33

1 How PVSCF works

1.1 Methodology of the VSCF approach

The definition of the true potential is

$$V(Q_1, Q_2, \dots, Q_N) = \sum_{i=1}^N V_i^{(i)}(Q_i) + V_c(Q_1, Q_2, \dots, Q_N), \quad (1)$$

where the total energy with MP1 correction is computed as,

$$E_{n_1 n_2 \dots n_N} = \sum_{i=1}^N \epsilon_{n_i} - (N-1) \left\langle \prod_{i=1}^N \phi_{n_i} \left| V_c(Q_1, Q_2, \dots, Q_N) \right| \prod_{i=1}^N \phi_{n_i} \right\rangle. \quad (2)$$

Where the total coupling potential $V_c(Q_1, Q_2, \dots, Q_N)$ is explicitly used. If another definition of the total potential is used, Equation (2) has to be modified. If a two mode limited representation of the coupling potential is used,

$$V_c(Q_1, Q_2, \dots, Q_N) = \sum_{i=1}^N \sum_{j>i} V_{ij}^{(2)}(Q_i Q_j), \quad (3)$$

Equation (2) can be simplified,

$$E_{n_1 n_2 \dots n_N} = \sum_{i=1}^N \epsilon_{n_i} - \left\langle \prod_{i=1}^N \phi_{n_i} \left| \sum_{i=1}^N \sum_{j>i} V_{ij}^{(2)}(Q_i Q_j) \right| \prod_{i=1}^N \phi_{n_i} \right\rangle \quad (4)$$

because the factor N has now been replaced by a N_c factor equal to 2. The value of N_c is related to the dimension of the expansion of the potential.

1.2 Implementation of the calculation of the total energy

To compute the MP1 energy correction the fact that,

$$\left\langle \prod_{i=1}^N \phi_{n_i} \left| V_c(Q_1, Q_2, \dots, Q_N) \right| \prod_{i=1}^N \phi_{n_i} \right\rangle = \left\langle \prod_{i=1}^N \phi_{n_i} \left| \sum_{i=1}^N \sum_{j>i} V_{ij}^{(2)}(Q_i Q_j) \right| \prod_{i=1}^N \phi_{n_i} \right\rangle \quad (5)$$

$$= \frac{1}{2} \left\langle \prod_{i=1}^N \phi_{n_i} \left| \sum_{i=1}^N \sum_{j \neq i} V_{ij}^{(2)}(Q_i Q_j) \right| \prod_{i=1}^N \phi_{n_i} \right\rangle \quad (6)$$

is used, where the factor $\frac{1}{2}$ avoids a double counting of each coupling term $V_{ij}^{(2)}(Q_i Q_j)$. Moreover, if the definition of the effective potential is used:

$$V_{eff}(Q_i) = V_i^{(1)}(Q_i) + \left\langle \prod_{j \neq i} \phi_{n_j} \left| \sum_{j \neq i} V_{ij}^{(2)}(Q_i Q_j) \right| \prod_{j \neq i} \phi_{n_j} \right\rangle. \quad (7)$$

Then the MP1 corrections can be rewritten as

$$\left\langle \prod_{i=1}^N \phi_{n_i} \left| V_c(Q_1, Q_2, \dots, Q_N) \right| \prod_{i=1}^N \phi_{n_i} \right\rangle = \frac{1}{2} \sum_{i=1}^N \left\langle \phi_{n_i} \left| V_{eff}(Q_i) - V_i^{(1)}(Q_i) \right| \phi_{n_i} \right\rangle. \quad (8)$$

This is the actual version used in the PVSCF program. Please note that when the total energy is computed, the shift which is multiplied by a factor $\frac{1}{2}$ is included. This multiplication comes from the fact that to compute the shift the sum of the contribution of each effective potential is used (for computing energy levels with). In this calculation there is a double counting of each coupling term $V(Q_i, Q_j)$, which is the reason why we had to multiply by a factor of $\frac{1}{2}$.

1.3 Vibrational excited states from perturbation theory

The standard perturbation theory limited at the second order is utilised to improve the VSCF energy of a reference state previously optimised. The corrected energy at the second order for the configuration state $|\Psi_{\mathbf{n}}\rangle$ is given by:

$$E_{\mathbf{n}}^{CC} = E_{\mathbf{n}}^{VSCF} + \sum_{\mathbf{m} \neq \mathbf{n}} \frac{|\langle \Psi_{\mathbf{n}} | \Delta V(Q_1, Q_2, \dots, Q_N) | \Psi_{\mathbf{m}} \rangle|^2}{E_{\mathbf{n}}^{(0)} - E_{\mathbf{m}}^{(0)}}. \quad (9)$$

- $E_X^{(0)}$ is the total energy at the zero order (solutions of the self-consistent eigenvalue problem) for the configuration state X.
- Each $|\Psi_{\mathbf{m}}\rangle$ state is a virtual excited state (obtained from the subroutine VIRTSTATE) for the reference VSCF state optimized $|\Psi_{\mathbf{n}}\rangle$.

All excited states are not used to compute MP2 corrections.^{1,2} The VGB (Vibrational Generalized Brillouin) theorem³ has been applied, which allows the matrix elements which involve a $|\Psi_{\mathbf{m}}\rangle$ state which is a monoexcitation of the reference state $|\Psi_{\mathbf{n}}\rangle$ to be ignored. This approximation is the same as the one used in Hartree–Fock and its extensions (MP2, MCSCF, ...). The perturbative potential operator of the VSCF Hamiltonian can be defined in the pairwise approximation by:

$$\Delta V(Q_1, Q_2, \dots, Q_N) = \sum_i \sum_{j>i} V^{(2)}(Q_i, Q_j) - \sum_{j=1}^N V_{eff}(Q_j). \quad (10)$$

If monoexcitations are not considered,^{1,3} then the matrix element of the perturbative operator for the numerator of Equation (9) is obtained:

$$\Delta V_{\mathbf{nm}} = \left\langle \prod_{i=1}^N \varphi_i^{(\mathbf{n})}(Q_i) \left| \sum_i \sum_{j>i} V^{(2)}(Q_i, Q_j) \right| \prod_{j=1}^N \varphi_j^{(\mathbf{m})}(Q_j) \right\rangle \quad (11)$$

$$= \sum_i \sum_{j>i} \int \int \varphi_j^{(\mathbf{n})}(Q_j) \varphi_i^{(\mathbf{n})}(Q_i) V^{(2)}(Q_i, Q_j) \varphi_j^{(\mathbf{m})}(Q_j) \varphi_i^{(\mathbf{m})}(Q_i) dQ_i dQ_j. \quad (12)$$

The matrix element $\Delta V_{\mathbf{nm}}$ of equation (12) was computed by evaluating each coupling term, $\Delta V_{\mathbf{nm}}^{(ij)}$, involved in the double sum, in two steps:

1. The integration for the Q_j coordinates (by means of the subroutine `INTEG1D`), allows for the computation of a one dimensional potential,

$$\bar{V}(Q_i) = \int \varphi_j^{(\mathbf{n})}(Q_j) V^{(2)}(Q_i, Q_j) \varphi_j^{(\mathbf{m})}(Q_j) dQ_j. \quad (13)$$

2. Integrate this new potential depending on the Q_i coordinates by computing,

$$\Delta V_{\mathbf{nm}}^{(ij)} = \int \varphi_i^{(\mathbf{n})}(Q_i) \bar{V}(Q_i) \varphi_i^{(\mathbf{m})}(Q_i) dQ_i. \quad (14)$$

1.4 Vibrational excited states from the interaction of configurations

A VCI approach is used to variationally compute a vibrational excited configuration optimized in VSCF. This method uses the virtual excited states built with the subroutine `VIRTSTATE` (including mono and bi-excitations this time²) and consists of a diagonalization of the total Hamiltonian,

$$\mathbf{H} = \mathbf{H}_0 + \Delta V(Q_1, Q_2, \dots, Q_N). \quad (15)$$

Where \mathbf{H}_0 is the VSCF Hamiltonian optimised for a given vibrational configuration. The perturbative operator can be defined by,

$$\Delta V(Q_1, Q_2, \dots, Q_N) = \sum_i \sum_{j>i} V^{(2)}(Q_i, Q_j) - \sum_{j=1}^N V_{eff}(Q_j). \quad (16)$$

Where the effective potential $V_{eff}(Q_j)$ contains the one diagonal potential $V^{(1)}(Q_j)$. The Hamiltonian matrix, which is symmetric, is computed by evaluating each matrix element $\langle \Psi_{\mathbf{n}} | \mathbf{H} | \Psi_{\mathbf{m}} \rangle$ defined as,

$$\langle \Psi_{\mathbf{n}} | \mathbf{H} | \Psi_{\mathbf{m}} \rangle = E_{\mathbf{n}}^{\text{VSCF}} \delta_{\mathbf{nm}} + \langle \Psi_{\mathbf{n}} | \Delta V | \Psi_{\mathbf{m}} \rangle. \quad (17)$$

Where the first term on the right side of Equation (17) is the VSCF energy (with an orthogonal basis set) and the second term is the coupling potential term between two vibrational configurations. This term can split into two different integral calculations with Equation (16):

1. Calculation of the pairwise coupling matrix element

$$\begin{aligned}\Delta V_{\mathbf{nm}} &= \left\langle \Psi_{\mathbf{n}} \left| \sum_i \sum_{j>i} V^{(2)}(Q_i, Q_j) \right| \Psi_{\mathbf{m}} \right\rangle \\ &= \sum_i \sum_{j>i} \int \int \varphi_j^{(\mathbf{n})}(Q_j) \varphi_i^{(\mathbf{n})}(Q_i) V^{(2)}(Q_i, Q_j) \varphi_j^{(\mathbf{m})}(Q_j) \varphi_i^{(\mathbf{m})}(Q_i) dQ_i dQ_j.\end{aligned}\quad (18)$$

Which is a sum over all pairwise coupling. Each term of this sum was obtained in two steps:

- (a) An integration for the Q_j coordinates (by means of the subroutine INTEG1D) is performed, allowing for the computation of a one dimensional potential,

$$\bar{V}(Q_i) = \int \varphi_j^{(\mathbf{n})}(Q_j) V^{(2)}(Q_i, Q_j) \varphi_j^{(\mathbf{m})}(Q_j) dQ_j.\quad (20)$$

- (b) An integration of this new potential, depending on the Q_i coordinates by computing,

$$\Delta V_{\mathbf{nm}}^{(ij)} = \int \varphi_i^{(\mathbf{n})}(Q_i) \bar{V}(Q_i) \varphi_i^{(\mathbf{m})}(Q_i) dQ_i.\quad (21)$$

2. Calculation of the effective potential matrix element

$$V_{\mathbf{nm}} = \left\langle \prod_{j=1}^N \varphi_j^{(\mathbf{n})}(Q_j) \left| \sum_j V_{eff}(Q_j) \right| \prod_{j=1}^N \varphi_j^{(\mathbf{m})}(Q_j) \right\rangle\quad (22)$$

$$= \sum_{j=1}^N \left\langle \varphi_j^{(\mathbf{n})}(Q_j) \left| \sum_j V_{eff}(Q_j) \right| \varphi_j^{(\mathbf{m})}(Q_j) \right\rangle \times \prod_{k \neq j} \delta_{n_k m_k}.\quad (23)$$

This is computed with the subroutine INTEG_ONEMODE. All effective potentials were previously computed before calling the VCI subroutine in VSCF.f90 because they are computed for only one vibrational configuration optimised in the VSCF program.

Finally the subtraction of the two contributions as it was written in Equation (10) is performed. For the diagonalization the direct method can be used if the basis set is not too large (subroutine DSYEV from BLAS library) or with iterative methods (subroutine DAVIDSON).

1.5 Calculation of harmonic frequency and anharmonicity constants

Assuming a truncated Dunham expansion to the fourth power,

$$E(\nu) = \omega_e \left(\nu + \frac{1}{2} \right) - \omega_e x_e \left(\nu + \frac{1}{2} \right)^2 + \omega_e y_e \left(\nu + \frac{1}{2} \right)^3 + \omega_e z_e \left(\nu + \frac{1}{2} \right)^4. \quad (24)$$

PVSCF will calculate the harmonic frequency, ω_e , along with the first three anharmonicity constants $\omega_e x_e$, $\omega_e y_e$ and $\omega_e z_e$. This is done by simply taking the vibrational energy levels and finding the differences between them, giving ΔG values. These values are used to calculate $\Delta^2 G$, $\Delta^3 G$ and $\Delta^4 G$ which are further differences in ΔG values, such that $\Delta^2 G$ is a difference between ΔG values, $\Delta^3 G$ is a difference between $\Delta^2 G$ values and so on.⁴

$$\Delta^4 G_{\nu+2} = \Delta^3 G_{\nu+\frac{5}{2}} - \Delta^3 G_{\nu+\frac{3}{2}} = 24\omega_e z_e \quad (25)$$

$$\Delta^3 G_{\nu+\frac{3}{2}} = \Delta^2 G_{\nu+2} - \Delta^2 G_{\nu+1} = (6\omega_e y_e + 36\omega_e z_e) + 24\omega_e z_e \left(\nu + \frac{1}{2} \right) \quad (26)$$

$$\begin{aligned} \Delta^2 G_{\nu+1} &= \Delta G_{\nu+\frac{3}{2}} - \Delta G_{\nu+\frac{1}{2}} \\ &= -(2\omega_e x_e - 6\omega_e y_e - 14\omega_e z_e) + (6\omega_e y_e + 24\omega_e z_e) \left(\nu + \frac{1}{2} \right) \\ &\quad + 12\omega_e z_e \left(\nu + \frac{1}{2} \right) \end{aligned} \quad (27)$$

$$\begin{aligned} \Delta G_{\nu+\frac{1}{2}} &= G(\nu+1) - G(\nu) = (\omega_e - \omega_e x_e + \omega_e y_e + \omega_e z_e) \\ &\quad - (2\omega_e x_e - 3\omega_e y_e - 4\omega_e z_e) \left(\nu + \frac{1}{2} \right) \\ &\quad + (3\omega_e y_e + 6\omega_e z_e) \left(\nu + \frac{1}{2} \right)^2 + 4\omega_e z_e \left(\nu + \frac{1}{2} \right)^3. \end{aligned} \quad (28)$$

Anharmonicity constants and the harmonic frequency are implemented for diatomic molecules only. Polyatomic molecules are to be implemented at a later date.

1.6 Construction of the Hessian matrix and projection of translations and rotations

1. Hessian construction:

The Hessian is constructed numerically via central differences. In the ACCURATE method, each second derivative is computed using four points (plus the equilibrium):⁵

$$\frac{\partial^2 f_{0,0}}{\partial x^2} = \frac{1}{12h^2} (-f_{2,0} + 16f_{1,0} - 30f_{0,0} + 16f_{-1,0} - f_{-2,0}) + O(h^4) \quad (29)$$

$$\frac{\partial^2 f_{0,0}}{\partial x \partial y} = \frac{1}{4h^2} (f_{1,1} - f_{1,-1} - f_{-1,1} + f_{-1,-1}) + O(h^2) \quad (30)$$

Here, h is the numerical step size, and $f_{x,y}$ is the value of the function at position (x, y) . Additionally, the methods **MEDIUM**⁶ and **SEMI**⁷ (using gradients) will be implemented.

2. Hessian projection⁸:

In order to remove the translations and rotations from the Hessian matrix, a transformation matrix \mathbf{T} has to be constructed. First, the center of mass \mathbf{r}_{COM} is calculated (with m_α mass and \mathbf{r}_α position of atom α):

$$\mathbf{r}_{COM} = \frac{\sum_{\alpha} m_{\alpha} \mathbf{r}_{\alpha}}{\sum_{\alpha} m_{\alpha}}; \quad (31)$$

the origin of the molecular coordinate system is shifted to the center of mass:

$$\mathbf{r}_{COM,\alpha} = \mathbf{r}_{\alpha} - \mathbf{r}_{COM}. \quad (32)$$

The moment of inertia tensor is constructed:

$$\mathbf{I} = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix}, \quad (33)$$

with

$$\begin{aligned} I_{xx} &= \sum_{\alpha} m_{\alpha} (y_{\alpha}^2 + z_{\alpha}^2) \\ I_{yy} &= \sum_{\alpha} m_{\alpha} (x_{\alpha}^2 + z_{\alpha}^2) \\ I_{zz} &= \sum_{\alpha} m_{\alpha} (x_{\alpha}^2 + y_{\alpha}^2) \\ I_{xy} = I_{yx} &= -\sum_{\alpha} m_{\alpha} (x_{\alpha} y_{\alpha}) \\ I_{xz} = I_{zx} &= -\sum_{\alpha} m_{\alpha} (x_{\alpha} z_{\alpha}) \\ I_{yz} = I_{zy} &= -\sum_{\alpha} m_{\alpha} (y_{\alpha} z_{\alpha}), \end{aligned} \quad (34)$$

where x_{α} , y_{α} , and z_{α} stand for the (mass-centered) x, y, and z-coordinate of atom α . The inertia tensor is diagonalized, yielding the principal moments of inertia (the eigenvalues). A zero eigenvalue indicates a linear molecule. The matrix \mathbf{D} formed of the eigenvectors of \mathbf{I} is used to generate the vectors of rotation (\mathbf{r}^1 , \mathbf{r}^2 , and \mathbf{r}^3):

$$\mathbf{d}_i = x_{\alpha} \mathbf{D}_{1,i} + y_{\alpha} \mathbf{D}_{2,i} + z_{\alpha} \mathbf{D}_{3,i}$$

$$\begin{aligned}
\mathbf{r}_{(3(\alpha-1)+i)}^1 &= (\mathbf{d}_2 \cdot \mathbf{D}_{i,3} - \mathbf{d}_3 \cdot \mathbf{D}_{i,2}) \cdot \sqrt{m_\alpha} \\
\mathbf{r}_{(3(\alpha-1)+i)}^2 &= (\mathbf{d}_3 \cdot \mathbf{D}_{i,1} - \mathbf{d}_1 \cdot \mathbf{D}_{i,3}) \cdot \sqrt{m_\alpha} \\
\mathbf{r}_{(3(\alpha-1)+i)}^3 &= (\mathbf{d}_1 \cdot \mathbf{D}_{i,2} - \mathbf{d}_2 \cdot \mathbf{D}_{i,1}) \cdot \sqrt{m_\alpha},
\end{aligned} \tag{35}$$

with $i = 1, 2, 3$ (for the x, y, and z-coordinate of atom α). The translational vectors are simply:

$$\begin{aligned}
\mathbf{t}_{(3(\alpha-1)+i)}^1 &= \mathbf{e}_{x,i} \cdot \sqrt{m_\alpha} \\
\mathbf{t}_{(3(\alpha-1)+i)}^2 &= \mathbf{e}_{y,i} \cdot \sqrt{m_\alpha} \\
\mathbf{t}_{(3(\alpha-1)+i)}^3 &= \mathbf{e}_{z,i} \cdot \sqrt{m_\alpha},
\end{aligned} \tag{36}$$

with $\mathbf{e}_{x,i}$ unit vector and $i = 1, 2, 3$. Via a Gram-Schmidt procedure, N_{vib} orthogonal vectors to the six (five) vectors of translation and rotation are generated and normalized. These vectors correspond to vibrations and are stored in the N_{vib} columns of the transformation matrix \mathbf{T} . The Hessian matrix is mass-weighted

$$\mathbf{H}_{mwc} = \frac{\mathbf{H}_{cart,\alpha,\beta}}{\sqrt{m_\alpha m_\beta}}, \tag{37}$$

then transformed from mass-weighted coordinates to internal coordinates by multiplication with \mathbf{T} and \mathbf{T}^\dagger :

$$\mathbf{H}_{int} = \mathbf{T}^\dagger \mathbf{H}_{mwc} \mathbf{T}. \tag{38}$$

The result is a $N_{vib} \times N_{vib}$ matrix, which is diagonalized. From the eigenvalues, the vibrational frequencies are calculated by the square root of the product of the eigenvalue and a factor from physical constants. The eigenvectors $\mathbf{q}_{int,j}$, which correspond to the normal modes, have to be transformed back from internal to cartesian coordinates $\mathbf{q}_{cart,j}$:

$$\mathbf{q}_{cart,j} = \frac{1}{\sqrt{m_\alpha}} \mathbf{T} \mathbf{q}_{int,j}, \tag{39}$$

with j indicating the mode number.

Number of rotations and translations: Several systems with a different number of vibrational modes can be distinguished.

- The system can consist of only one atom; then, the translations are treated as vibrations, and no Hessian purification is possible.
- If the system contains two atoms, the number of normal modes is set to $3N - 5$, and a Hessian purification is possible.

- The system is periodic: Purification can be carried out and removes only the translations, and $3N - 3$ vibrational modes are obtained.
- A partial Hessian cannot be purified, and yields $3N_{\text{active}}$ normal modes.
- The partial Hessian is an *embedded* Hessian (e.g., a molecule in a liquid): The translations and rotations of the *active* part of the system are removed.
- At any other system, the moment of inertia tensor is calculated, irrespective of the purification. The number of normal modes is set to $3N - 6$ / $3N - 5$, depending on the eigenvalues of the moment of inertia tensor.

Why the partial Hessian should not be purified:

- Only part of the system was moved to generate the Hessian matrix, the other part was kept fixed. The partial Hessian should not be contaminated by overall translations and rotations. Additionally, the translation and rotation of the active part of the system changes the distance to the inactive fragment.
- Removing the rotations and translations introduces a large number of artificial modes, because they are removed from a part of the system which did not move initially. The artificial modes complicate a treatment via VSCF.

2 Getting started with and using PVSCF

2.1 Introduction to PVSCF

PVSCF requires a number of sections which define what the program does. Within each section there are options to refine the calculation. All sections must have start with a \$ and have an end (\$END). Not all sections are necessary for every calculation and can thus be omitted when not needed. Available sections include:

- **RUN** Gives the overall type of calculation to be run.
- **SYSTEM** Provides information on the system.
- **RESTART** Used for restarting a calculation.
- **HESSIAN** For calculating a Hessian.
- **MAKEPES** Contains options necessary to generate a potential energy surface.
- **SOLVER** Defines the method for calculating the eigenvalues and eigenvectors.

- **PES** Provides PVSCF with a/the potential/s generated from the creation of a PES. This section also includes the minimum energy for rescaling the PES. This section has a subsection called **MULTIPES**.
- **POSTVSCF** This section tells PVSCF what type of advanced vibrational calculations to do e.g., VCI etc. .
- **TDVCI** For TDVCI calculations.
- **FVSCF** Used for FAST-VSCF calculations.
- **SAVEWF** Used for saving the wavefunction and effective potentials.

Within the RUN section the command **RUNTYPE** specifies the various types of computations:

- **MAKEPES** Activates the potential energy surface generation routine.
- **HESSIAN** A Hessian matrix is constructed numerically.
- **DIAG** Performs a diagonal calculation on a pre-computed potential energy surface.
- **VSCF** Performs a VSCF calculation on a pre-computed potential energy surface.

Within the SYSTEM section there are the following options:

- **NMODES** This specifies the number of modes present. Remember to adjust based on the geometry of your system (linear vs non-linear).
- **XYZFILE** Defines the `.xyz` file needed in conjunction with some *ab initio* programs. These files are generated by the respective *ab initio* programs. For a full list see Section (2.12).
- **XVFILE** Defines the `.XV` file needed in conjunction with SIESTA. This file is an output file from SIESTA.
- **LINEAR** Used to specify if the molecule is **LINEAR**, the default is non-linear. Issues can arise if your system is not perfectly linear and you define it as thus.
- **PERIODIC** Used if the system is periodic.
- **PARTIAL** If this keyword is present, the partial Hessian construction is activated; no purification is possible.
- **PURIFY** Translations and rotations are removed. This is only valid for isolated systems as the system is not translationally/rotationally invariant. The default is no Hessian purification.

- **EMBEDDED** The translations and rotations of the active part of the system are removed from the partial Hessian.
- **TNUM** Is used when working in curvilinear coordinates (when reading in data from TNUM). Speak to one of the David's for details.

Within the RESTART section has two options:

- **RESTART_FILE** Tells PVSCF to use a restart file that must be provided.
- **TIME_INTERVAL**

The SOLVER section has a number of options:

- **SOLVER_TYPE** Has two sub options, **FGH** Defines Fourier grid Hamiltonian as the solver. **SGH** Defines SGH as the solver.
- **GTENSOR_ORDER** This option is still under development and pertains to the use of curvilinear coordinates. Currently a only a zero value is supported.
- **NODES** Number of nodes for the interpolation.
- **CONVERGENCE** *Used in VSCF.f90 for MP1 correction???*
- **POTENTIAL_TOL** *Used in VSCF.f90 for MP1 correction???*
- **MIXING** *Used for mixing potentials??*

The HESSIAN section has a number of useful options:

- **METHOD** Method selection: **ACCURATE** is the only option available currently and selected by default; **SEMI** (for semi-numerical second derivatives) and **MEDIUM** are planned.
- **MEDIUM** *Under development.*
- **ACCURATE**
- **SEMI**
- **DISPL** Numerical step size in Bohrs (0.01 per default).

The MAKEPES section has many options:

- **HESSIAN_FILE** The name of the file containing the Hessian and molecular data.

- **ABINITIO_PACKAGE** Specifies which *ab initio* program was used to compute the Hessian. At present only GAMESS-US, CPMD, CP2K (use: RUN_TYPE VIBRATIONAL_ANALYSIS and PRINT_LEVEL MEDIUM in &GLOBAL section), and VASP (requires OUTCAR and .xyz-file with coordinates in Ångstrom) are allowed. SIESTA has been added recently and requires a .XV file. If the Hessian was constructed with PVSCF, the option NONE has to be set here).
- **PES_CALCULATION** Describes which type of potential energy surface is to be computed (at present only 1D, 2D or ALL where all equals both 1D and 2D are allowed).
- **INTERFACE** Defines the type of interfacing used between PVSCF and the *ab initio* program (at present only DIRECT, BATCH or NUMBERED are allowed).
- **ENERGY_SCRIPT** If the DIRECT option is used, this entry defines the name of the script file used to compute the energy of the configurations produced by PVSCF. Note that PVSCF will generate a file named: DIRECT1D.inp for 1D potential energy surfaces and DIRECT2D.inp for 2D PES, launch the energy script and wait until it has finished. It is assumed that the energy script takes care of transforming the data produced by the *ab initio* program into a meaningful PES for PVSCF.
- **TEMPLATE_FILE** Specifies the name of the template file containing the description of an input file for the *ab initio* program used (so far either CPMD or GAMESS-US, others are possible by making your own.) (see also section 2.4 for details on how to create a template file). If the file does not exist, the program will look for my_template.tpl but will exit if this file is not present either.
- **SCAN_RANGE** This keyword enables the user to specify the maximum distance, *in units of normal-mode coordinates*, used for constructing the PES scan. This leads to the generation of distortion coordinates along the normal modes ranging from $Q_{\min}^i = -\frac{\text{scan_range}}{\sqrt{\omega_i}}$ to $Q_{\max}^i = +\frac{\text{scan_range}}{\sqrt{\omega_i}}$. The default value is 4.0.
- **SCAN_POINTS** This option specifies the number of points, n_{points} , used to compute the PES curve. This also influences the size of the PES scan steps, dQ_i , since they are given by: $dQ_i = \frac{2 \cdot Q_{\max}^i}{n_{\text{points}}}$. The default value is 16 points.
- **DO_ONLY_1D** This option can be used to restart a MAKEPES in order to compute the 1D potential energy surface for only a few modes. The next line has to contain a list of the requested modes (using PVSCF numbering scheme, 1 is the lowest frequency mode and $3N - 6$ or $3N - 5$ are the highest frequency modes).
- **QPOINTS_OPT** This keyword enables an optimisation of the position of the grid points, see section 2.10 for details.
- **MASS** This keyword allows the user to change the mass of any given atoms (referred to by their sequence number in the Hessian or coordinate file) in order to compute the

normal modes of isotopically substituted compounds. This keyword is followed by the number of substituted atoms on a single line and then the number of the atom(s) and mass in atomic mass units. The following example is used for deuterated water (oxygen is the first atom followed by two hydrogens with adjusted mass.):

```
$MAKEPES
...
MASS
2
2 2.014
3 2.014
$END
```

The PES section has the following options:

- **PES_1D** Defines the 1D potential. This is calculated by MAKEPES but you need to tell PVSCF where it is. The default name for this file is `new_1D_pot`.
- **PES_2D** Defines the 2D potential. The default name for this file is `new_2D_pot`.
- **PES_3D** Defines the 3D potential. *This is not implemented yet.*
- **ENERGY** Defines the energy at the minimum. This is set as the new zero. To get this value you need to run (with the program of your choice) the file `MINIMUM.inp` which is an output after making a PES.
- **DIPOLE** Used for defining the dipole. PVSCF expects three values, x, y and z for the dipole.
- **RESCALE_PES** This tells PVSCF which potentials to rescale and has numerous sub options. **NONE** Results in no rescaling. **1D** Rescales the 1D potential. **2D** Rescales the 2D potential. **3D** *This is not supported at this time.* **ALL** Both the 1D and the 2D potentials are rescaled.
- **INTERFACE** Determines how the generation of input files will be handled. This has numerous sub options: **DIRECT** Computes each point directly, **SCAN** puts multiple points in a single input file, **MD** does MD scans of the PES, **BATCH** generates all input files in one go, **NUMBERED** generates some numbered input files (the calculation is controlled by the script) and **MODEWISE** generates all input files for the 1D PES and all couplings to the same mode in one go.
- **MODE_DEPSCAN** This option can be used to adjust the scan range for each mode from the default. PVSCF will suggest a better scan range in the output (of a vibrational calculation) which can be used to rescan with. The number of calculated points can also

be changed, if desired (the default of 16 works most of the time though and more points can mean significantly longer calculations).

- **MINI_LOC** *Helps to see if you are actually at the minimum???*

The MULTIPES section (a subsection of PES), used for mixing potentials has the following options:

- **HIGH_PES_1D** For defining a higher quality 1D PES
- **HIGH_ENERGY** For defining the minimum energy of the **HIGH_PES_1D**.
- **HIGH_DIPOLE** For defining the dipole of the **HIGH_PES_1D**.

The VSCF section has the following options:

- **SA_VSCF** *No idea what this does???*
- **SCALING_FACTOR_MIN** ???
- **SCALING_FACTOR_MAX** ???
- **SCALING_FACTOR_STEP** ???
- **LOCALISE_EXCITED** Localisation constraint on the excited states.
- **OVERTONE_LIST**
- **COMBINATION_LIST**

The POSTVSCF section can be used for the vibrational analysis and has a plethora of options:

- **CORRTYP** Type of post-VSCF calculation requested (at present only MP2, CI or ALL are allowed).
- **NMAX_MP2** Specifies the maximum number of quanta used for the MP2 calculations (the default value is 4). This is very dependent on the quality of your scan and therefore the number of bound states present.
- **NMAX_CI** Specifies the maximum excitation quanta used for the generation of configurations in the CI basis (the default value is 2). Again this is very dependent on the quality of your scan.
- **AUTO_MAX_QUANTA** This option is necessary to perform VCI or CC-VSCF calculations in a physical vibrational basis. The maximum value of the excited quanta for one given mode is automatically chosen in order to have the latest vibrational eigenvalue lower than the extreme *ab initio* grid points of the PES.

- **FGHVCI** Specifies the use of [Fourier??] grid Hamiltonian (FGH) method to build the VCI Hamiltonian matrix or to evaluate the direct action of the Hamiltonian in the Direct Davidson scheme.
- **VCI_LEVEL** Can be used for computing VCI eigenvalue with the use of triple and/or quadruple excitations. By default, the value is 2 to take into account single+double excitations (VCISD).
- **REDUCED_BASIS** Specifies the use of a reduced VCI basis set which the size is limited by considering all states for which the sum of excited quanta of the virtual VSCF states is equal to NMAX_CI.
- **VCIPSI** Can be used to perform VCI calculations in a very small active space. It uses an iterative VCI scheme with a perturbative criteria of the excited states which had to be added in the active space. The diagonalisation is realised with the Davidson algorithm.
- **VCIPSI_LAPACK** Same option as the **VCIPSI** but where the diagonalisation is realised with LAPACK library.
- **EIGVCI_PT2** Is an option which is used when the VCIPSI option is in use to correct the converged eigenvalue by PT2 theory (to consider the latest excited states of the inactive space).
- **STOREBIMODE** Specifies that all one-mode integrals and two mode integrals are stored in memory for VCI calculations (highly recommended for DIRECT option, see below).
- **GSVCI** Triggers a ground-state only VCI calculation (see below for details).
- **GDMP2**
- **DCMP2**
- **INIT_P0_VSCF**
- **TOLCIPSI**
- **TOLPT2**
- **LIM**
- **EIGVCI_PT2**
- **STOREBIMODE**
- **CI_DIAG** with sub options of **DAVIDSON**, **LAPACK** and **DIRECT**.

The section TDVCI contains the following options:

- **RESTART** Restarts the TDVCI from the last step.
- **PIT** Starts a propagation in imaginary time.
- **TFIN**
- **10.0** Final propagation time.

The section FVSCF, used for FAST-VSCF has the following options:

- **SNGALL** Requests a single-to-all calculation for the modes specified in the next line.
- **FGRID** {PAIR or NxN} Requests a FAST-algorithm using only the mode-mode couplings given below. The **PAIR** notation needs a series of mode mode pairs (e.g., 12 11), each given on a separate line. The **NxN** notation needs a list of $N(N - 1)/2$ numbers (1 for coupled and 0 for not coupled), each given on a separate line. NOTE: This notation starts with the higher frequency first, so it goes from N to 1 to ensure compatibility with the existing FAST-VSCF implementation in GAMESS.

The section SAVEWF, used for saving the wavefunction and effective potentials has the following options:

- **DIAG**
- **CONVWF**
- **TRACE**
- **2DPOT**

2.2 Generating a potential energy surface

The generation of potential energy surface is activated by using:

```
$RUN
  RUNTYPE
  MAKEPES
$END
```

Followed by a \$MAKEPES . . . \$END section:

```
$MAKEPES
  HESSIAN_FILE
```

```

MOLVIB
ABINITIO_PACKAGE
CPMD
PES_CALCULATION
1D
INTERFACE
DIRECT
ENERGY_SCRIPT
my_energy_script2
TEMPLATE_FILE
my_template.tpl
SCAN_RANGE
4.0
SCAN_POINTS
16
QPOINTS_OPT
$END

```

2.3 Diagonal frequencies

The calculation of 1D diagonal frequencies is a basic step in PVSCF. It is crucial to carry out this step to validate your 1D potential **before** moving onto any calculations of a 2D potential and VSCF. An example input is given:

```

$RUN
RUNTYPE
DIAG
$END

$SYSTEM
NMODES
  an integer number goes here
PURIFY
$END

$MAKEPES
HESSIAN_FILE
  filename.hess
ABINITIO_PACKAGE
ORCA
PES_CALCULATION

```

```

1D
INTERFACE
  NUMBERED
ENERGY_SCRIPT
  my_energy_script2
TEMPLATE_FILE
  my_template.tmpl
$END

$SOLVER
SOLVER_TYPE
  FGH
NODES
  256
$END

$PES
PES_1D
  new_1D_pot
ENERGY
  minimum energy value goes here
RESCALE_PES
  1D
$END

```

Provided that you have all the correct script/template files this will work.

2.4 Template files

The template files use a simple tagging language to identify where PVSCF needs to put the relevant data. The user needs to provide a template file that can perform a single point energy calculation with the desired *ab initio* program. In order to generate the necessary potential energy surfaces, PVSCF will create the input files by using the template, inserting the coordinates of the system adding a line that describe the sequence number and the position of the point on the potential energy surfaces where appropriate. The available tags are:

- The single tag <INFOTAG> is replaced by the sequence number and PES location information. Note that this element is necessary for the proper construction of the PES.
- The section <COORDS> describes the layout of the atomic coordinates. This section has to be terminated with the <ENDCOORDS> tag to be valid. Within this section, the following tags are available:

- <ATNAME> is replaced by the corresponding atom name.
- <ATNUM> is replaced by the atomic number of the corresponding element (mainly used for GAMESS-US and SIESTA inputs).
- <XPOSA> x coordinate of the atom in Ångstroms.
- <YPOSA> y coordinate of the atom in Ångstroms.
- <ZPOSA> z coordinate of the atom in Ångstroms.
- <XPOSB> x coordinate of the atom in atomic units (bohrs).
- <YPOSB> y coordinate of the atom in atomic units (bohrs).
- <ZPOSB> z coordinate of the atom in atomic units (bohrs).

The coordinate tags are usually combined together so that an input file that requires coordinates in the following format:

```
$COORDINATES
C  1.00 5.00 2.77
...
H  2.00 1.00 6.22
$END
```

is described in the template file by:

```
$COORDINATES
<COORDS>
<ATNAME> <XPOSA> <YPOSA> <ZPOSA>
<ENDCOORDS>
$END
```

You only need a single line to represent the whole set of atomic positions.

Some programs, e.g., CPMD require that each atomic species is defined separately. The coordinate section of the input file looks like this:

```
&ATOMS
*O_SG_pade_q6 KLEINMAN-BYLANDER
LMAX=P
  2
  3.000 4.000 6.222
  3.560 1.000 7.333
```

```
*C_SG_pade_q4 KLEINMAN-BYLANDER
LMAX=P
  1
0.300 0.100 2.400
&END
```

This can be achieved by repeating the <COORDS> section for each type of atom in order to tell PVSCF where to place the coordinates. Note that each <COORDS> keyword has to be followed by the name of the required element (i.e., for gold use <COORDS> Au):

```
&ATOMS
*O_SG_pade_q6 KLEINMAN-BYLANDER
LMAX=P
  2
<COORDS> O
<XPOSB> <YPOSB> <ZPOSB>
<ENDCOORDS>

*C_SG_pade_q4 KLEINMAN-BYLANDER
LMAX=P
  1
<COORDS> C
<XPOSB> <YPOSB> <ZPOSB>
<ENDCOORDS>
&END
```

Remember that PVSCF will not check that all atomic species are being written, so check that you have a set of coordinate tags for each one in the system.

Finally, note that there are currently no options to allow PVSCF to complete partial lines. This means that the tags given in the template files will always be replaced in full and remove whatever was put on the same line.

2.5 Hessian calculation

A Hessian calculation is performed in two steps: First, numerical displacements of the atoms are generated and calculated with an *ab initio* program. All molecular information, energy values etc., are written to a Hessian file (see my_hessian.template.tmpl). In the second step, the Hessian file is read and the Hessian matrix is constructed numerically.

A. For generating the displacements, the keyword `HESSIAN` is required in the `$RUN ... $END` section. A `$HESSIAN ... $END` section has to be present with the keywords:

PARTIAL A partial Hessian will be constructed. This requires an additional column in the `.xyz` file after the z-coordinate: 1 if atom is included, 0 if not. The default is a full Hessian calculation.

METHOD Method selection: `ACCURATE` is the only option available currently and selected by default; `SEMI` (for semi-numerical second derivatives) and `MEDIUM` are planned.

DISPL Numerical step size in Bohr (0.01 per default).

The energy script for the *ab initio* calculation has to be given in the `$MAKEPES ... $END` section (`ENERGY_SCRIPT`), as well as a name for the Hessian file (`HESSIAN_FILE`), and an *ab initio* template file (`TEMPLATE_FILE`). In the `$SYSTEM ... $END` section, the input geometry in Ångstrom has to be given as `.xyz` file:

```
$RUN
  RUNTYPE
  HESSIAN
$END

$SYSTEM
  NMODES
  18
  XYZFILE
  ethane.xyz
$END

$MAKEPES
  HESSIAN_FILE
  ethane_hessian
  ABINITIO_PACKAGE
  NONE
  PES_CALCULATION
  1D
  INTERFACE
  DIRECT
  ENERGY_SCRIPT
  my_energy_script_for_hessian
  TEMPLATE_FILE
  my_template_for_hessian.tmpl
$END
```

```
$HESSIAN
  PARTIAL
  ACCURATE
  DISPL
  0.01
$END
```

B. For constructing the Hessian matrix, the keyword `MAKEPES` is required in the `$RUN ... $END` section; in the `$MAKEPES` section, a Hessian file name must be given (containing the energy values from the displacements), and the `ABINITIO_PACKAGE` keyword `NONE` must be set. The PES calculation will directly start after printing the Hessian matrix; if that is not desired, the *ab initio* template file should not be present.

```
$RUN
  RUNTYPE
  MAKEPES
$END
```

```
$SYSTEM
  NMODES
  18
  XYZFILE
  ethane.xyz
  PARTIAL/PERIODIC/EMBEDDED
  PURIFY
$END
```

```
$MAKEPES
  HESSIAN_FILE
  ethane_hessian
  ABINITIO_PACKAGE
  NONE
  PES_CALCULATION
  1D
  INTERFACE
  DIRECT
  ENERGY_SCRIPT
  my_energy_script_for_pes
  TEMPLATE_FILE
  my_template_for_pes.tpl
$END
```


2.6 Post-VSCF calculations

The calculation of post-VSCF corrections is triggered by the presence of a `$POSTVSCF ... $END` section. This section contains, for example:

```
$POSTVSCF
  CORRTP
  MP2
  NMAX_MP2
  6
$END
```

If you choose the CI option, the diagonalisation of the VCI matrix is realized by default with the Davidson iterative algorithm (subroutine `DAVIDSON` from the `ITERATIVE` module). If we want to use the direct diagonalisation scheme (subroutine `DSYEV` of the `LAPACK` library), you have to add the following section before the final `$END` of the input file:

```
$CI_DIAG
  LAPACK
$END
```

When you use the `DAVIDSON` algorithm to perform diagonalisation, the cutoff of the norm of the residue is 1 cm^{-1} by default. If you want to change this value you have to add the following section before the final `$END` of the input file:

```
$CI_DIAG
  TOLRES_DAVIDSON
  0.01
$END
```

and the Davidson diagonalisation will be performed with a cutoff of the norm of the residue is 0.01 cm^{-1} .

If the storage of the VCI Hamiltonian matrix is not possible (too high memory access), it is possible to use a Direct Davidson scheme which will not build the matrix but only use the evaluation of the Hamiltonian on a virtual configuration. To select this option you have to use the following section before the final `$END` of the input file:

```
$CI_DIAG
  DIRECT
$END
```

When this option is used it is recorded to equally select the option STOREBIMODE to perform calculations in a reasonable time.

From version 0.35 (27 November 2008), PVSCF can also perform ground-state only VCI calculations. This is similar to the technique used in MULTIMODE, where the vibrational ground state is optimised using VSCF and then the computed VSCF virtual states are used to perform a VCI calculation. Note that for strongly coupled systems, this approach is usually less accurate than the VSCF/VCI scheme that PVSCF uses by default. The presence of the keyword GSVCI in the \$POSTVSCF ... \$END section triggers the calculation, which only requires 1 VSCF and 1 VCI run and thus could be attractive for large systems. Please note that this scheme needs *all* eigenvalues and eigenvectors and is thus restricted to using LAPACK (default) or other full incore diagonalisation methods. The iterative Davidson approach (incore or direct) is not compatible with GSVCI.

2.7 TDVCI

TDVCI calculations are initiated by the keyword TDVCI in the \$RUN ... \$END section:

```
$RUN
  RUNTYPE
  TDVCI
$END
```

Further TDVCI options can be specified in the \$TDVCI ... \$END section:

```
$TDVCI
  RESTART
$END
```

2.8 FAST-VSCF methods

As of the middle of February 2007, PVSCF has an experimental implementation of the Fast-VCI method. This is a currently non-tested algorithm that is designed to speed up vibrational CI (VCI) calculations on large systems. So far the implementation is fairly basic: the keyword FASTVSCF has to be present in the \$RUN ... \$END section and a \$FVSCF ... \$END section has to be present to avoid any unpredictable behaviour. The \$FVSCF ... \$END section contains, for example:

```
$FVSCF
  SNGALL
  12
$END
```

This would trigger a SINGLE-TO-ALL calculation for the mode 12 only.

2.8.1 What is implemented?

At present a screening of the $\langle \varphi_{n_j} \varphi_{n_i} | V^{(2)}(Q_i, Q_j) | \varphi_{m_i} \varphi_{m_j} \rangle$ CI integrals is implemented, along with a screening of the effective potential contributions,

$$\left\langle \prod_{j \neq i}^n \phi_j^{k_j}(Q_j) \middle| V(Q_1, \dots, Q_n) \middle| \prod_{j \neq i}^n \phi_j^{k_j}(Q_j) \right\rangle \quad (40)$$

(veffective.f90 routines). This means that the VSCF procedure is faster when using FAST-VSCF and that CI calculations also run noticeably quicker. Preliminary tests on methanol show a 6-fold speedup for each VCI calculation when the OH stretch is active (coupled to all modes) and the rest of the modes are diagonal. The FAST algorithm is now also implemented in the MP2 corrections. As of 18/05/2007, the presence of the FAST-VSCF keyword in the \$RUN ... \$END section (along with accompanying \$FVSCF ... \$END section) is also taken into account during MAKEPES runs. This enables the user to compute only the parts of the 2D coupling potential necessary for a FAST-VSCF calculation. Reading in of a PES generated during a FAST-VSCF run of MAKEPES is now ok.

2.9 Saving wavefunctions

The computed diagonal wavefunction, converged VSCF wavefunction and effective potentials can be saved to disk in ASCII format by using the \$SAVEWF ... \$END section. This section contains, for example:

```
$SAVEWF
  DIAG
  CONVWF
  TRACE
  2DPOT
$END
```

2.10 Optimisation of PES sampling

As of beginning of January 2008, it is now possible to use PVSCF to optimise the position of the grid points used to explore the PES. The method is based on a strict PO-DVR criteria (more details to come....)(It's been 7 years? is more detail coming or not?). This is a three step procedure which needs to be run in the following sequence:

1. Run a standard 1D PES scan using a relatively large range (for example `SCAN_RANGE 8.0` and `SCAN_POINTS 32` or `16`)
2. Using the generated 1D curves, run a diagonal calculation with the keyword `QPOINTS_OPT` in the `$MAKEPES` section along with the relevant `SCAN_RANGE`. This will generate 12 optimised points for each normal mode and store them in a file called “Q-OPT-POINTS”
3. Now you can run a PES scan using the optimised grid points contained in the “Q-OPT-POINTS” file by specifying the keyword `QPOINTS_OPT` in the `$MAKEPES` section. Note that your new number of scan points is automatically adjusted to 12.

The accuracy of the new scheme has been tested on the water molecule and the method generates a difference $\leq 2 \text{ cm}^{-1}$ at the VCI level compared to a regular grid of 16 points.

Note that the PES scanning range has been reduced in the new GAMESS-US VSCF implementation (Version 11 April 2008). Their range now corresponds to setting `SCAN_RANGE 3.87298` in PVSCF.

2.11 Degenerate perturbation theory

In version 0.29, a degenerate perturbation theory treatment for MP2 calculations was introduced. The simple approach described by Daněček and Bouř⁹ which accounts for 2-level degeneracies was followed. Note that this approach was first introduced Truhlar and coworkers for VPT2¹⁰ and for electronic MP2 calculations.¹¹ In order to avoid the problems encountered with the second-order energy correction in Equation (9), that is:

$$\begin{aligned}
 E_{\mathbf{n}}^{\text{CC}} &= E_{\mathbf{n}}^{\text{VSCF}} + \sum_{\mathbf{m} \neq \mathbf{n}} \frac{|\langle \Psi_{\mathbf{n}} | \Delta V(Q_1, Q_2, \dots, Q_N) | \Psi_{\mathbf{m}} \rangle|^2}{E_{\mathbf{n}}^{(0)} - E_{\mathbf{m}}^{(0)}} \\
 &= E_{\mathbf{n}}^{\text{VSCF}} + \underbrace{\sum_{\mathbf{m} \neq \mathbf{n}} \frac{W_{nm}}{E_{\mathbf{n}}^{(0)} - E_{\mathbf{m}}^{(0)}}}_{E_n^{(2)}}
 \end{aligned} \tag{41}$$

when $E_{\mathbf{n}}^{(0)} - E_{\mathbf{m}}^{(0)} \rightarrow 0$, the expression of $E_n^{(2)}$ is replaced by:

$$E_{\mathbf{n}}^{(2)} = \frac{1}{2} \sum_{\mathbf{m} \neq \mathbf{n}} \left[E_{\mathbf{m}}^{(0)} - E_{\mathbf{n}}^{(0)} \pm \sqrt{\left(E_{\mathbf{m}}^{(0)} - E_{\mathbf{n}}^{(0)} \right)^2 + 4|W_{nm}|^2} \right] \tag{42}$$

where the + signs holds for $E_{\mathbf{n}}^{(0)} > E_{\mathbf{m}}^{(0)}$ and the – sign for $E_{\mathbf{n}}^{(0)} \leq E_{\mathbf{m}}^{(0)}$. In their original paper, Daněček and Bouř define $E_n = \sum \epsilon_{n_i} - \langle \Psi | V_c | \Psi \rangle$, which in our notation is in fact $E_{\mathbf{n}}^{\text{VSCF}}$.

Equation (42) is the solution of N two-level problems of the type:

$$\begin{vmatrix} E_{\mathbf{n}}^{(0)} & \langle n | \Delta V | m \rangle \\ \langle m | \Delta V | n \rangle & E_{\mathbf{m}}^{(0)} \end{vmatrix} = 0 \tag{43}$$

We see that if both levels are non-degenerate and $E_n^{(0)} < E_m^{(0)}$, we obtain for the energy of the lower state:

$$E_n^{(2)} = \frac{1}{2} \left[E_m^{(0)} - E_n^{(0)} - \sqrt{\left(E_m^{(0)} - E_n^{(0)} \right)^2 + 4|W_{nm}|^2} \right] \quad (44)$$

This method is triggered by the presence of the keyword DCMP2 in the \$POSTVSCF section.

2.12 Interfacing with *ab initio* programs

2.12.1 Orca

In the current version of Orca, use the following header to obtain a reliable frequency calculation. For example, a PBE0 calculation (optimisation followed by Hessian) is given by:

```
!RKS PBE0 VeryTightSCF TightOpt Grid6 NumFreq
%freq CentralDiff true # use central-differences
      Increment 0.005 # increment in bohr for the
                       # differentiation (default 0.005)
end
```

Moreover, PVSCF **ALWAYS** projects translations and rotations when using Orca Hessians (use PURIFY in \$SYSTEM).

2.12.2 GAMESS-US

In GAMESS-US, it is possible to reduce the amount of printout to a minimum (i.e., no orbitals, etc.) by using NPRINT=-5. Calculating a PES in an automated way (as PVSCF does) requires that every point is converged as much as possible. This is usually achieved for the HF wavefunction by using the following SCF line:

```
$SCF DIRSCF=.t. DIIS=.t. SHIFT=.t. ETHRSH=1 SWDIIS=0.00001 DEMCUT=0.1
      SOGTOL=0.005 RSTRCT=.f. EXTRAP=.t. DEM=.f. DAMP=.t. FDIFF=.t. $END
```

This has been optimised somewhat over the years and seems to give the most reliable results. The options RSTRCT=.f. DEM=.f. are not required as these are the default but it is usually a good idea to have them there in case of problems.

2.12.3 NWChem

Currently, the only way to get NWChem to work on xgrid is to specify the basis set *directly* in the template file.

2.12.4 CPMD

For optimisations of an *isolated* molecule, use a large cell and:

```
&CPMD
OPTIMIZE GEOMETRY XYZ
LBFGS
HESSIAN UNIT
CONVERGENCE ORBITALS
1E-7
CONVERGENCE GEOMETRY
5E-5
ISOLATED MOLECULE
&END
```

Then for the subsequent Hessian calculation:

```
&CPMD
VIBRATIONAL ANALYSIS GAUSS
CONVERGENCE ORBITALS
1E-7
ISOLATED MOLECULE
RESTART WAVEFUNCTION COORDINATES LATEST
&END
```

Note that you can also use the LR option of Vibrational Analysis to shave a few seconds off your calculation in some cases (most of the time FD is more general and faster).

2.12.5 CP2K

When doing frequency calculations, DO NOT USE:

```
&SUBSYS
&TOPOLOGY
CENTER_COORDINATES
&END
```

This generates several artificial imaginary frequencies. However, it is fine to use it when doing geometry or wavefunction optimisations.

For metals, the OT method should not be used. It is necessary to add MOs to the SCF procedure, and perform a smearing of the occupation numbers. The SCF section Joerg [Do you mean Jürg?] Hutter suggested in the CP2K Google Group is:

```
&SCF
  SCF_GUESS RESTART
  EPS_SCF 5.0E-7
  MAX_SCF 500
  ADDED_MOS 500
  CHOLESKY INVERSE
  &SMEAR ON
    METHOD FERMI_DIRAC
    ELECTRONIC_TEMPERATURE [K] 300
  &END SMEAR
  &DIAGONALIZATION
    ALGORITHM STANDARD
  &END DIAGONALIZATION
  &MIXING
    METHOD BROYDEN_MIXING
    ALPHA 0.1
    BETA 1.5
    NBROYDEN 8
  &END
  &PRINT
    &RESTART
    &EACH
      QS_SCF 50
    &END
    ADD_LAST NUMERIC
  &END
&END SCF
```

Mixing is also essential for the wavefunction convergence.

Also, the CP2K developers suggested not playing around with `EPS_DEFAULT` and `DENSITY_CUTOFF` – at least not using larger values than the defaults!

2.12.6 PSI3

PSI3 can sometimes get stuck on the wrong state during scans without a reliable “guess” for the SCF wavefunction. The best technique so far is to run the file MINIMUM.inp first (or as soon as possible after the calculation has started) and use the temp file (xxx.32) as a restart for the calculation of the other points (remember to use C1 symmetry throughout with `subgroup=c1` in the `psi` section). This can be achieved by specifying `chkpt_mos=true` in the `psi` section and not forgetting to specify the correct temporary file (with the `-p` option). Also note that PSI3 uses `/tmp` to store temporary files so having the same tempfile name for all calculations is not a good idea.

To avoid jobs getting stuck it might be safer to use the direct SCF routines by inserting the following section in the input file:

```
scf: (
  direct = true
)
```

On top of that, reducing cache levels seems to improve the job’s stability on multi-core machines. Therefore the `psi` section should contain the line: `cachelev = 0`.

2.12.7 DALTON2

Dalton needs two files to be able to start a calculation (even if the manual says it can run with one file...): a `.dal` file and a `.mol` file. PVSCF will generate the `.mol` file, based on the template, but the user needs to provide a DALFILE [`.dal` file??] which will contain the DALTON commands. This file is encoded by the xgrid script. There are also some problems with basis set: xgrid cannot read them from the default location so they have to be copied into the script or in the `.mol` files. Note that the Hückel guess in DALTON needs the “ano-4” basis! This basis has to be provided in order for any calculations to run smoothly. For some CC runs (either ground state or EOM-CC) the scan will need more iterations to converge (micro-iterations in DALTON). After having had a loong look at the code, it seems that the variable to set is `MAX IT` in the `*CC INPUT` section:

```
*CC INPUT
.CC2
.FREEZE
4 0
.MAX IT
150
*CCEXCI
.NCCEXCI
3
```


2.12.8 SIESTA

SIESTA produces the force constant matrix (`example.FC`) and PVSCF constructs the Hessian based on this. The atomic positions are also read from SIESTA's equivalent of an xyz file (`example.XV`) so that displacements can be made in the vibrational analysis. SIESTA struggles to converge wavefunctions to a very desirable criteria (10^{-7} eV). It is suggested that 10^{-4} eV is used as the convergence criteria to start with, tests with your system should be performed to see what accuracy SIESTA can provide. (There is a way of compiling SIESTA to get double precision throughout but the standard as of version 3.2 is that some parts remain single precision which might be a cause of considerable inaccuracy).

Be careful with filtering (`FilterCutoff` or `FilterTol`) when convergence testing for cutoff (Fix the filtering for the convergence test). Generally cutoff in SIESTA is not the same as in CPMD/CP2K. `MeshCutoff` is more like a traditional cutoff but you need to beware of the filtering that accompanies it; you have been warned. Filtering can be disabled if required.

2.13 Curvilinear coordinate VSCF

This part of PVSCF is still under development and requires an extra section that describes the curvilinear coordinate system. (contact either David's about it). In order to use generic curvilinear coordinates, the metric tensor is generated during the PES generation through the use of the `TNUM` keyword in the `SYSTEM` section. The tensor is save in the file `G-TENSOR` and is necessary for the subsequent VSCF calculation.

At the minute, only the zeroth-order approximation to the full curvilinear kinetic Hamiltonian is implemented and requires the presence of the section

```
GTENSOR_ORDER
0
```

in the `SOLVER` section.

2.13.1 Curvilinear coordinates: a practical guide

Starting from the optimised geometry (corresponding to the computed Hessian), load it into `macmolpt` and change the coordinates to z-matrix. Copy the z-matrix coordinates into a file. Now use the `zmat_to_tnum` script (usage: `zmat_to_tnum filename`) to generate the `tnum_zmat` file. This file can now be copied into the `SYSTEM` section after:

```
TNUM
&geom zmat=T NM_TO_sym=t nat=29 sym=F /
```

where `nat` is the number of atoms. Don't forget to add a blank line before the `$END` of the section.

PVSCF/`tnum` will usually complain that the geometry isn't quite what it hoped for and will print out a more "suitable" set of coordinates (in GAMESS-US format, WARNING: these are in BOHRS). Re-run the Hessian calculation using these new coordinates, without re-orienting them, and you should then be able to run PVSCF/`tnum` with the new Hessian.

3 PVSCF usage

3.1 VCI calculations

New VSCF protocol for VCI calculations: First perform a 1D pes calculation, followed by a diagonal run. Check that all modes have enough bound states (at least 8 for a VCI calculation). If they have less than 8, increase `QRANGE` and the number of sampling points until you obtain the required numbers of bound states. Finally, run the PES calculation using the new range. Note that the number of sampling points can be reduced using the DVR `QPOINTS` optimise routine.

Still to implement: Mode dependent `Qrange` and sampling points.

3.2 VMP2

DO NOT USE: Auto max quanta for MP2 calculations, without checking that you have at least 7 bound states!

When using the FAST-VSCF approximation, it is recommended that `DCMP2` is used instead of standard VMP2 to avoid problems with accidental degeneracies.

3.3 Constants

As of 30/12/2008, PVSCF uses the fundamental constants definitions of CODATA 2006 (see <http://www.codata.org>). Note that this might lead to some small changes in the computed harmonic frequencies.

References

- [1] L. Pele, B. Brauer and R.B. Gerber, *TheoChemAcc* (2006).
- [2] L.S. Norris, M.A. Ratner, A.E. Roitberg and R.B. Gerber, *J. Chem. Phys.*, **105** (1996) 11261.

- [3] F. Culot and J. Lievin, *TheoChemAcc*, (**89**), 227 (1994).
- [4] G. Herzberg, *Molecular spectra and molecular structure Second Edition*, Van Nostrand Reihold company, 1950.
- [5] *Numerical Analysis*, pp. 884, eq. 25.3.24 and 25.3.26.
- [6] *Numerical Analysis*, pp. 884, eq. 25.3.23 and 25.3.27.
- [7] *Numerical Analysis*, pp. 883, eq. 25.3.21 and pp .884, eq. 25.3.26.
- [8] Gaussian white paper “Vibrational Analysis in Gaussian”, J.W. Ochterski, Oct. 1999.
- [9] P. Daněček and P. Bouř, *J. Comp. Chem.*, **28** (2007) 1617.
- [10] K.M. Kuhler, D.G. Truhlar and A.D. Isaacson, *J. Chem. Phys.*, **104** (1996) 4664.
- [11] X. Assfeld, J.E. Almlöf and D.G. Truhlar *Chem. Phys. Lett.*, **241** (1995) 438.