# Xiamen Valence Bond

## An *ab initio* Non-Orthogonal Valence Bond Program

Version 2.0

**Lingchun Song, Zhenhua Chen, Fuming Ying, Jinshuai Song, Xun Chen, Peifeng Su, Yirong Mo, Qianer Zhang, Wei Wu***

*State Key Laboratory of Physical Chemistry of Solid Surfaces,*

*Fujian Provincial Key Laboratory of Theoretical and Computational Chemistry,*

*and Department of Chemistry,*

*Xiamen University, Xiamen, Fujian 361005, CHINA*

*weiwu@xmu.edu.cn*

Latest updated on Jun 22, 2012

# Referencing for XMVB

## A) J. Chem. Phys. format

The *ab initio* Valence Bond calculations are performed with the XMVB program.[1]

1. (a) L. Song, Z. Chen, F. Ying, J. Song, X. Chen, P. Su, Y. Mo, Q. Zhang, W. Wu, XMVB 2.0: An *ab initio* Non-orthogonal Valence Bond Program, Xiamen University, Xiamen 361005, China, 2012. (b) L. Song, Y. Mo, Q. Zhang, W. Wu, J. Comput. Chem. 2005, 26, 514.

## B) American Chemical Society format

The *ab initio* Valence Bond calculations are performed with the XMVB program.[1]

1. (a) Song, L.; Chen, Z.; Ying, F.; Song, J.; Chen, X.; Su, P.; Mo, Y.; Zhang, Q.; Wu W., XMVB 2.0: An *ab initio* Non-Orthogonal Valence Bond Program, Xiamen University, Xiamen 361005, China, 2012. (b) Song, L.; Mo, Y.; Zhang, Q.; Wu, W. *J. Comput. Chem.* **2005**, *26*, 514.

# Availability of XMVB

XMVB version 2.0 is a stand-alone program that performs *ab initio* valence bond calculations. XMVB version 2.0 will be distributed from Xiamen University for free. The principal user will be asked to fill a license agreement and send the signed agreement to:

Professor Wei Wu
Department of Chemistry
Xiamen University,
Xiamen, Fujian 361005
P. R. China
Tel: 86-592-2182825 Fax: 86-592-2184708
E-mail: weiwu@xmu.edu.cn

# CONTENTS

# 1. INTRODUCTION

## 1.1 What is XMVB?

Xiamen Valence Bond (XMVB) is a quantum chemistry program for performing electronic structure calculations based on the non-orthogonal Valence Bond methods.

In the current arena of theoretical and computational chemistry, molecular orbital (MO) based methods undoubtedly are enjoying the overwhelming dominance, largely due to their high computational efficiencies. Apart from the MO theory, however, valence bond (VB) theory remains the widespread conceptual matrix for chemists, thus the development of novel algorithms and practical programs for the *ab initio* VB theory have been an endeavor for many of us.

The VB project at Xiamen University was initiated in 1986, when a spin-free form of VB method was independently proposed. In 1989, we wrote a simple VB code and applied it to $H_3$ molecule. The systematic development of a complete and efficient VB code nevertheless gained the momentum in 1992, when an algorithm based on the left coset decomposition of the symmetric group was proposed and programmed.[1,2] In 1995, we further developed an algorithm called the paired-permanent-determinant (PPD) approach,[3] which is more efficient for systems of many covalent bonds than the traditional Slater expansion algorithm, on which our previous VB code, Xiamen-99, was based. In the last twenty years,, we continued optimizing the code and implementing some newly developed VB approaches into the code. To distinguish our VB code from our city whose name is Xiamen, the code was renamed as XMVB[4] in 2002, beginning with Version 1.0. After ten years, Version 2.0 is now released.

The motivation of the development of the XMVB package is to provide chemists with a well-designed and efficient program to perform both classical and modern valence bond calculations at *ab initio* level. To gain chemical lucidity from VB theory, the XMVB package uses Heitler-London-Slater–Pauling (HLSP) functions as state functions. The superposition of a set of HLSP functions result in the wave function for a system. VB orbitals may be strictly localized, delocalized, or bonded-distorted (semi-localized), depending on particular applications.

The capacity of XMVB is limited by the system memory. By default, the maximum number of active orbitals and active electrons are both 18; the maximum number of VB structures for VBCI is 25000; the maximum number of electrons is 100, and the maximum number of basis functions and orbitals are both 200. This consumes about 30GB memory.

Currently, there are 2 mechanisms of distribution available for users: module and stand-alone package. The module distribution is a module for the GAMESS-US package; while the stand-alone distribution obtains integrals from utility PREINT and performs VB computations as a stand-alone program. The syntaxes of XMVB input files are the same for both distributions. Incorporating our VB code into the GAMESS-US package allows more hybrid VB methods, such as DFVB, VBSCF(CAS), VBPCM, and VBEFP, to be performed. Thus, we strongly recommend users to choose the module distribution.

## 1.2 What's New in Version 2.0?

The following features are available in XMVB 2.0:

- A new module distribution combined with GAMESS-US is officially released.
- A new algorithm for orbital optimization with analytical gradients is available.
- New VB methods, including VB Perturbation Theory (VBPT2) and Density Functional Valence Bond (DFVB) are implemented.
- Ab inito VB methods with solvation effects, including VBPCM, VBEFP, and VBEFP/PCM, are implemented by incorporating with GAMESS-US package.
- Block Localized Wavefunction (BLW) is available in XMVB 2.0.
- Population analysis, including Mulliken and Löwdin atomic charges, atomic spin density polarization, Mayer's bond order and dipole moments can be calculated in XMVB 2.0.
- More new keywords are designated.
- WFN file for AIM2000 with VB density now can be generated easily by incorporating with GAMESS-US package.

## 1.3 What can be done with XMVB?

- Following ab initio VB methods are implemented in XMVB:

  Valence Bond Self-Consistent Field (VBSCF);

  Breathing Orbital Valence Bond (BOVB);

  Valence Bond Configuration Interaction (VBCI);

  Valence Bond Perturbation Theory (VBPT2);

  Density Functional Valence Bond (DFVB);

  Valence Bond Polarizable Continuum Model (VBPCM);

  Valence Bond Effective Fragment Potential (VBEFP);

  Combined Valence Bond Effective Fragment Potential Polarizable Continuum Model (VBEFP/PCM).
- Many-electron wave function of computed system is written in terms of HLSP functions or Slater determinants. Both of structural coefficients and weights are provided.
- The form of VB orbitals is flexible; they may be HAOs, BDOs, or OEOs, which are strictly localized, semi-localized, or fully delocalized.
- Following population analysis properties of wave function are computed:

  Mulliken and Löwdin atomic charges;

  Atomic spin density polarization;

Mayer’s bond order;

Dipole moments.

- Interface files with other programs, such as AIM2000 and Moden, are provided. Thus, more properties may be computed, and VB orbitals may be views by third-party programs.

# 2. INSTALLATION

Both distributions of XMVB are currently available for LINUX platform. 1.5GB RAM is required. Followings are the instructions for installation. Note that the source code will NEVER be released to the users. Only compiled object file or executable files are available for users.

## 2.1 For Module Distribution

After obtaining the compiled object xmvb.o of XMVB program, one will need to modify

```
set XMVB=false
```

to

```
set XMVB=true
```

in the lked script in GAMESS-US package. Then run the lked script to get GAMESS executable with the whole capability of XMVB.

## 2.2 For Stand-Alone Distribution

The stand-alone distribution is released as a package of compiled executable files. To install the stand-alone distribution, the users should:

A. Unpack the compressed tar file by using the following command,

```
tar xvfz xmvb.tar.gz
```

Once the file is unpacked successfully, a new directory xmvb/ will be created.

B. Set the VBDIR environment variable to the location of XMVB package and append xmvb/ to your $PATH. The variable VBDIR is essential for PPD algorithm and utility PREINT(see Section 6.2).

# 3. RUNNING A JOB

An XMVB job consists of the following two steps:

*Step 1.* Prepare the integrals of primitive basis functions and the MO orbitals of the system. An input file including molecular information and basis set information is needed. Files “1e.tem”, “2e.tem” and “orb.mo” will be generated after this step.

*Step 2.* Run XMVB calculation. A file with extension name “.xmi”(XMI file, see chapter 4) is needed here.

The programs, utilities and files needed in these 2 steps differ in different distributions. The details are shown below.

## 3.1 For Module Distribution

The two steps of an XMVB job in module distribution are done in one shot with command

```
rungms job [VERNO] [1]
```

after all files are prepared.

*Step 1* is done in GAMESS program with a GAMESS input file including line “$CONTRL VBTYP=XMVB $END”. XMVB calculation will be proceeded automatically following the completion of *step 1*.

## 3.2 For Stand-Alone Distribution

With stand-alone distribution used, *step 1* is done with program PREINT(see Section 6.2). After running PREINT with a proper input file, “1e.tem”, “2e.tem”, and “orb.mo” will be generated for XMVB. Type the following command to run XMVB job:

```
xmvb xmifile
```

where xmifile is the input file for XMVB.

# 4. INPUT FILE

## 4.1 Syntax of XMI File

An ".xmi" input file describes the information of the XMVB computation, containing up to 6 sections as "CTRL", "BFI", "STR", "FRAG", "ORB", "AIM", and "GUS". The following are the format and example of "xmi" file:

**$CTRL**
nmul=*val*1, nstr=*val*2, norb=*val*3, ...
**$END**
**$STR**
ntstr(1,1), ntstr(2,1), …, ntstr(nel,1)
⁞ ntstr(1,n), ntstr(2,n), …, ntstr(nel,n)
**$END**
**$BFI**
nfroz   nbas
List of frozen MOs
List of  basis functions
**$END**
**$FRAG**
*nf*(1), *nf*(2), …, *nf*(*N*)
*lf*(1,1), *lf*(2,1),…, *lf*(*nf*(1),1)
…
*lf*(1,*n*), *lf*(2,*n*),…, *lf*(*nf*(*n*),*n*)
**$END**
**$ORB**
*max*(1), *max*(2), …, *max*(*val*3)
*nvic*(1,1), *nvic*(2,1),…, *nvic*(*max*(1),1)
…
*nvic*(1,*n*), *nvic*(2,*n*),…, *nvic*(*max*(*n*),*n*)
**$END**
**$SYM**
n1 b1 n2 b2
…

**$END**
**$AIM**
WFN file name
**$END**
**$GUS**
VBO(1) MO for VBO(1)
…
VBO(*n*) MO for VBO(*n*)
**$END**

A minimal example of XMVB input file is shown below:

```
 VBSCF/3-21G  Molecule: HF
$ctrl
nstr=3
$end
$str
1 1 2 3  ; 3 structures
1 1 2 2
1 1 3 3
$end
$orb
4 4 2
3-6
4 3 5 6
1 2
$end
```

NOTE FOR INPUT:

*a.* There shall not be more than 400 characters in a line.

*b.* The first line is the explanation line.

*c.* Anything after “;” or “#” in a line will be regarded as comment. If they appear in the same line, the first appearing will be the dominance.

*d.* Section starts with $CTRL, $BFI, $STR, $FRAG, $ORB, $SYM, $AIM, $GUS and end with $END. Don’t write anything else in begin or end line.

*e.* All sections except $CTRL can be optional based on a user’s choice of keywords.

*f.* Sections can be placed in any order.

*g.* Abbreviation is available in the whole input file. For example, in the above example, string “4 4 2” may be written equivalently as “4*2 2”; “1 1 2 2” may be written equivalently as “1:2”; “3-6” means “3 4 5 6”.

## 4.2 CTRL Section

The $CTRL section contains the information of how a job is performed. The input format is **name**=*value* or **name**=*option*, except for the keywords which need no values or options. <enter> and <space> are used to separate keywords. If a keyword accepts several options in a time, the options are separated with “,”.

### 4.2.1 Keywords for Global Control

**BPREP:**

This keyword initiates an integral transformation from primitive basis functions to VB basis functions with $BFI (see below) needed. The transformation may freeze core orbitals, remove some primitive basis functions which are not involved in VB calculation, and hybridize basis functions. XMVB will use primitive basis functions without transformation if this keyword is missing.

NOTE:

This keyword cannot be used together with ORBTYP=HAO or GUESS=MO (see below).

**EPG=*n*:**

Set the convergent criterion of energy to $10^{-n}$. Default value is 7.

**ITMAX=*n*:**

*n* is the maximum number of iterations. Default value is 200.

**NMUL=*n*:**

*n* is the spin multiplicity ($2S+1$) of system. Default value is 1, which means singlet state.

**NAO=*m*:**

*m* is the number of active VB orbitals whose occupation number varies in the structures. NAO is required if keywords STR or ISCF=5 (see below) is specified.

**NAE=*n*:**

*n* is the number of active VB electrons which occupy the active orbitals. NAE is required if keywords STR or ISCF=5 (see below) is specified.

**NSTR=*n*:**

*n* is the number of VB structures (or determinants). This keyword can be omitted if STR(see below) is assigned.

**STR=*options*:**

This keyword generates VB structures automatically and hence NSTR and the $STRU section are not needed. This keyword requires NAO and NAE (see above) to declare the active space. Users may use one or several of the following options:

COV: Covalent structures will be generated.

ION[(*n*-*m*)]: Ionic structures will be generated. Simple ION will generate all ionic structures; ION(n) will generate only the *n-th* order ionic structures and ION(n-m) will generate ionic structures from the *n-th* to the *m-th* order.

FULL: All VB structures will be generated.

By default, VB structures will not be generated automatically and NSTR and $STRU are essential and neither NAO nor NAE is necessary.

**FIXC:**

Request to fix structure coefficients for VB structures. In VB theory, the coefficients are obtained by solving the secular equation **HC**=**SC**E. For some special purposes, one may want to fix the coefficients. The coefficients are inputted following the corresponding VB structures. For example, the following input will constrain the coefficients of the three VB structures to be 1.0:0.5:0.5

```
$STR
1 2  1.0
1 1  0.5
2 2  0.5
$END
```

**NSTATE=*n*:**

Energy, coefficients and weights of structures for the *n*-th excited state, rather than for the ground state, will be calculated and printed out. The values of *n* can be:

0: The ground state.(Default)

*n*: The *n*-th excited state.

NOTE:

*a.* Though calculation is asked for an excited state, VB orbitals are optimized by minimizing the ground state energy. When the *n*-th excited state is requested, the (*n*+1)-th root will be chosen as the *n*-th excited state when solving the secular equation. Thus, *n* must be smaller than the number of structures.

*b.* For VBCI calculaitons, NSTATE can be only 0 or 1.

**IPRINT=*n*:**

Printing levels for XMVB. The available levels are:

1: Initial guess, energy, coefficients, weights, and orbitals will be printed. This is the default printing level.

2: All contents in IPRINT=1, Hamiltonian and overlap matrices in terms of VB structures, and population analysis will be printed.

3: All contents in IPRINT=2, density matrix and orbital overlap matrix will be printed.

**CTOL=*tol*:**

Set the Coefficient TOLerance when printing coefficients and weights of VB structures. Only the coefficients and weights of VB structures whose absolute values of coefficients are not smaller than tolerance *tol* will be printed. The default tolerance is 0, which means all structures will be printed.

NOTE:

The tolerance *tol* is a real parameter. For instance,

**ctol=0.01**

It means that only structures whose absolute values of coefficients larger than or equal to 0.01 will be printed. For VBCI this keyword is not functioning.

**CICUT=*n*:**

Set cut threshold to $10^{-n}$ for CI configurations. The contribution of a CI configuration is estimated by perturbation theory. If the contribution is less than the threshold, the configuration will be discarded. This will reduce the computational effort for CI calculations. Recommended values are 5 or 6. Default value is 0 (no cut).

**NCOR=*m*:**

The first *m* orbitals (2*m* electrons) will be frozen in the VBCI or VBPT2 calculation. The default value is 0, which means all orbitals will be counted in VBCI or VBPT2.

**GUESS=*option*:**

This keyword describes the way to generate or read the initial guess for a VB computation. Valid options can be:

AUTO: The program automatically provides guess orbitals by diagonalizing a fragmant-localized Fock matrix. This is the default option.

UNIT: The first basis function of an orbital in $ORB is set to be the guess for the orbital.

READ: Guess orbitals are read from external file, which should be provided by user.

MO: Initial guess of VB orbitals will be obtained directly from molecular orbitals.

RDCI: Initial guess in VBCI type will be given by users.

NOTE:

*a.* GUESS=MO cannot be used with BPREP.

*b.* GUESS=AUTO cannot be used when some orbitals contain only one basis function (see section 4.6).

**WFNTYP=*option*:**

Options for the way to expand the many-electron wave functions of system.

STR: VB structures are used. (Default)

DET: VB determinants are used for state functions, instead of VB structures.

**VBFTYP=*option*:**

Options for the way to expand VB structures.

PPD: paired-permanent-determinant algorithm is used.

DET: Slater determinant algorithm is used.

By default, the program will decide which one to use according to the system, method, or algorithm the users choose.

NOTE:

1. PPD expansion can be used only with ISCF=1 or ISCF=3.
2. ISCF=5, VBPT2, VBCI, DFVB, solvation VB methods, DEN, and IPRINT⩾2 will use DET expansion automatically.
3. All systems with multiplicity larger than 2 will be calculated with DET expansion.
4. Systems with electrons in VB calculation larger than 14 will be calculated with DET expansion.

**ORBTYP=*option*:**

Specify the type of VB orbitals.

HAO: Hybrid Atomic Orbitals are used.

BDO: Bond Distorted Orbitals are used.

OEO: Overlap Enhanced Orbitals are used.

GEN: VB orbitals are defined by users. (Default)

NOTE:

*a.* $FRAG (see Section 4.5) is needed if ORBTYP=HAO is specified. The $FRAG section will specify the fragments based on atoms or basis functions and orbitals will be assigned in $ORB section based on the fragment definitions in $FRAG.

*b.* ORBTYP=OEO does not need $FRAG and $ORB sections since the OEOs are delocalized in the whole system.

*c.* ORBTYP=GEN does not need $FRAG section, and orbitals will be described in terms of VB basis functions explicitly in $ORB section.

*d.* ORBTYP=HAO cannot be used with BPREP.

*e.* ORBTYP=BDO can be used with other orbital types, such as ORBTYP=HAO as ORBTYP=HAO,BDO. ORBTYP=BDO is equivalent to ORBTYP=GEN,BDO.

**FRGTYP=*option*:**

Specify the type of fragments when ORBTYP=HAO.

| | |
|---|---|
| ATOM: | The fragments of system will be defined with atoms. This is the default. |
| SAO: | The fragments of system will be defined with symmetrized atomic orbitals. |

NOTE:

$FRAG (see Section 4.5) is required if FRGTYP=SAO. For FRGTYP=ATOM, each atom is considered as a fragment if no FRAG section appears in the input file.

### 4.2.2 Keywords for Computational Methods and Algorithms

**VBSCF:**

A VB Self-Consistent Field computation is requested. This is the default method for the XMVB program.

**BOVB:**

Ask for a Breathing Orbital VB (BOVB) calculation.

NOTE:

*a.* BOVB method cannot be used with VBCI.

*b.* BOVB method is usually more difficult to converge than VBSCF. Thus, it is recommended to run a BOVB job with a good initial guess. It is recommended to run a VBSCF calculation first, followed by the BOVB calculation with optimized VBSCF orbitals as the initial guess.

**BLW:**

Block Localized Wavefunction (BLW) method is requested. With this keyword specified, $STRU will not be read and the structure will be generated automatically. The users only need specify the type of VB orbitals (see FRGTYP and ORBTYP above).

NOTE:

The implementation of the BLW method in the program is not optimized. Users are recommended to run GAMESS-BLW for a BLW calculation.

**VBCIS:**

Ask for a VBCIS calculation.

**VBCISD:**

Ask for a VBCISD calculation.

**VBCIDS:**

Ask for a VBCIDS calculation.

**VBPT2:**

A VBPT2 computation will be performed.

**DFVB:**

Ask for a DFVB calculation.

**SCC:**

Size-Consistent Correction in DFVB computations will be switched on.

**VBEFP:**

Ask for a VBEFP calculation.

**VBPCM:**

Ask for a VBPCM calculation.

**VBEFPPCM:**

Ask for a VBEFP/PCM calculation.

**ISCF=*n*:**

ISCF specifies orbital optimization algorithm. The value *n* currently can be:

1: Numerical gradients with forward-difference approximation are used with the DFP-BFS algorithm.

2: Analytical gradients in terms of basis functions with the L-BFGS algorithm. This algorithm involves only the first-order density matrix and is not suitable for all systems.

3: Numerical gradients with central-difference approximation are used with the DFP-BFS algorithm.

4: Optimization with Generalized Brillium Theory(GBT) is requested.

5: Analytical gradients in terms of VB orbitals with the L-BFGS algorithm. This is the most efficient algorithm so far . This is the default option of XMVB.

### 4.2.3 Keywords for Wave Function Analysis

**BOYS:**

Boys localization is requested for the final VB orbitals.

NOTE:

*a.* It is strongly recommended to use this keyword for VBSCF. This makes VB orbitals easier to be interpreted and more physically meaningful.

*b.* Boys localization is available only for VBSCF method.

*c.* Boys localization can be only used in cases in which orbitals are separated into blocks, and there is no common basis function between blocks.

**DEN:**

First-order density matrix is requested. The result will be written to a file with extended name “den”.

**OUTPUT=AIM:**

WFN file for AIM2000 program will be printed. This is available only in module distribution. A $AIM with WFN filename is relevant for this keyword. Without $AIM, the content of WFN file will be stored in .dat file of GAMESS-US.

### 4.2.4 Keywords for previous version

The following keywords appear in the previous version and are not available in version 2.0. This part is important only for those who are used to the previous version.

**CIG:**

This keyword has been modified as GUESS=RDCI.

**DET:**

This keyword has been replaced by WFNTYP=DET.

**EXC:**

This keyword has been replaced by NSTATE=1.

**IOPT=*n*:**

This keyword has been replaced by keyword ISCF.

**IOUT=*n*:**

**NODIIS:**

**RHF:**

**VBXM=*n*:**

This keyword has been replaced by keywords WFNTYP and STRTYP.

## 4.3 BFI Section (Required for BPREP)

The BFI section specifies how to transform primitive basis function to VB basis functions. The primitive basis functions are those used in GAMESS, Gaussian or PREINT and VB basis functions are used in XMVB. The Syntax of $BFI section is:

```
$BFI
NFROZ   NBAS
List of frozen MOs
```

```
List of basis functions
$END
```

Here NFROZ is the number of frozen MOs and NBAS is the number of VB basis functions used in XMVB. Then frozen MOs and basis functions will be listed respectively. If there is no MO to be frozen, place a blank line there. The VB basis functions may be reordered according to how users list them. This new order will be used in ORB section for the orbital description.

Following are two examples for the BFI section. The first example comes from the HF molecule with 6-31G basis set, where basis functions are not hybridized:

```
$BFI
3 6
1 4 5
1 2 4 7 8 11
$END
```

In this example, three MOs 1, 4 and 5 are frozen and 6 VB basis functions are kept for the XMVB calculation. Primitive basis functions 3, 5, 6, 9 and 10 are removed from the list as the corresponding MOs are frozen. Note that the fourth VB basis function is the primitive basis function 7 according to the list, not primitive basis function 4 anymore.

The second example comes from the $CH_4$ molecule with 6-31G basis set, showing the orbital freezing and the hybridization of basis functions:

```
$BFI
3 8
1 3 4
1 1 1 1 1 1 3 3
1.0 1
1.0 2
1.0 4
1.0 7
1.0 8
1.0 11
0.5 12 0.5 14 0.5 16
0.5 13 0.5 15 0.5 17
 $END
```

Here MOs 1, 3 and 4 are frozen and 8 hybrid VB basis functions are used in XMVB calculation. Line “1 1 1 1 1 1 3 3” specifies the number of primitive basis function in each VB basis function. Following lines specifies how the VB basis functions are hybridized. In this example, the 7th VB basis function is a hybrid basis function composed of 3 primitive basis functions 12, 14 and 16.

## 4.4 STR Section

The STR section describes the information of VB structures or VB determinants if DET of CTRL section is specified. For VB structures, paired electrons, which may be lone pairs or covalent bonds, should be written first followed by unpaired electrons. The number of unpaired electrons depends on the spin multiplicity. For example: For a structure with three lone pairs (orbitals 1, 2, and 3), one covalent bond (orbitals 4 and 5), and one unpaired electron (orbital 6), the structure is expressed as,

1 1 2 2 3 3 4 5 6

For determinants, all alpha orbitals are listed first, followed by beta orbitals. For example: A determinant of alpha orbtials 1, 2, 3, 4, and 6 and beta orbtials 1, 2, 3, and 5 is expressed as

1 2 3 4 6 1 2 3 5

Note: It is strongly recommended to write the most important structure as the first one. This can avoid potential problems in VBCI.

If BOVB is specified in $CTRL section, the program will try to convert the VB orbitals into breathing orbitals. It uses automatically different orbitals for different structures. For example: If the initial VB structures are:

1 1 2 3

1 1 2 4

1 1 3 5

The program will convert them to:

1 1 2 3

6 6 7 4

8 8 9 5

Note that the VB structures should be independent. VB structures are recommended to be written in the following orders:

**Inactive Active**

where “Inactive” stands for the inactive orbitals which keep doubly occupied in all structures; “active” stands for the active orbitals whose occupation varies in the structures. The singly occupied orbitals in high-spin systems should always be put in the tail of the structures.

Following are the examples of typical bonding patterns and their corresponding $CTRL and $STRU section examples, in which only active orbitals are labeled:

**A. System of 2-electrons on 2-centers**

| | | |
|---|---|---|
| **A** ·—· **B** | **A**: $^{-}$ **B** $^{+}$ | **A** $^{+}$ **B**: $^{-}$ |
| **S1** | **S2** | **S3** |

$CTRL

```
nstr=3 nmul=1
$END
$STR
1 2  ; S1
1 1  ; S2
2 2  ; S3
$END
```

**B. System of 3-electrons on 2-centers**

**$^-$A:  ·B          A·  :B$^-$**

**S1                S2**

```
$CTRL
nstr=2 nmul=2
$END
$STR
1 1 2  ; S1
2 2 1  ; S2
$END
```

**C. System of 3-electrons on 3-centers**

**H · —— · H    · H          H ·    H· —— · H**

**S1                S2**

**H:$^-$    H$^+$    · H          H ·    H$^+$    :H $^-$**

**S3                S4**

**H$^+$    :H $^-$    · H          H ·    H:$^-$    H$^+$**

**S5                S6**

**H:$^-$    H ·    H$^+$          H$^+$    H ·    :H $^-$**

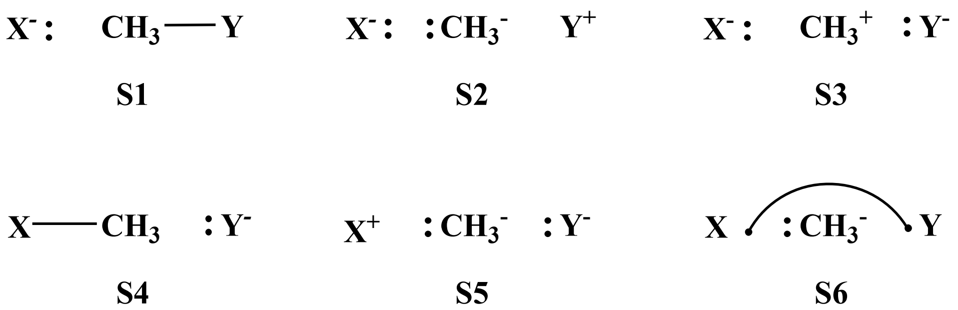
**S7                S8**

```
$CTRL
nstr=8 nmul=2
$END
$STR
1 2 3  ; S1
2 3 1  ; S2
```

```
1 1 3  ; S3
3 3 1  ; S4
2 2 3  ; S5
2 2 1  ; S6
1 1 2  ; S7
3 3 2  ; S8
$END
```

**D. System of 4-electrons and 3-centers**

6 VB structures (3 VB orbitals with 4 electrons, singlet)



```
$CTRL
nstr=6 nmul=1
$END
$STR
1 1 2 3  ; S1
1 1 2 2  ; S2
1 1 3 3  ; S3
1 2 3 3  ; S4
2 2 3 3  ; S5
2 2 1 3  ; S6
$END
```

## 4.5 FRAG Section (Relevant when ORBTYP=HAO)

Generally, the FRAG section is required if ORBTYP=HAO. In this section, fragments in which VB orbitals are localized will be defined and the orbitals will be generated with the basis functions specified in the fragments.

The syntax of $FRAG is:

$FRAG

nf(1), nf(2), … nf(*N*)

[basis function description(1)] *lf*(1,1), *lf*(2,1), ... *lf*(*nf*(1),1)

[basis function description(2)] *lf*(1,2), *lf*(2,2), ... *lf*(*nf*(2),2)
…
[basis function description(*N*)] *lf*(1,*N*), *lf*(2,*N*), ... *lf*(*nf*(*N*),*N*)
$END

Here the system is separated into *N* fragments. *nf*(*i*) means the number of atoms or basis functions in the *i*-th fragment, and *lf*(*j*,*i*) is the atom or basis function *j* in the *i*-th fragment. Basis function description is needed only when FRGTYP=SAO is chosen. Following is an example of $H_2$ molecule with FRGTYP=ATOM:

```
$CTRL
NSTR=3 ORBTYP=HAO
FRGTYP=ATOM
$END
$STRU
1 2
1 1
2 2
$END
$FRAG
1 1
1
2
$END
$ORB
1 1
1
2
$END
```

The above $FRAG specifies two fragments, where one atom is in each fragment. Fragment 1 includes the first H atom and fragment 2 includes the second H atom. With this definition, users only need to specify fragment in which an orbital is located in $ORB section. With FRGTYP=SAO, the fragments are specified by the type of basis functions. Following is an example of HF molecule with 6-31G basis set:

```
$CTRL
NSTR=3 VBFTYP=DET DEN
ISCF=5 NAO=2 NAE=2
ORBTYP=HAO FRGTYP=SAO
$END
$STRU
```

```
1:4 5 6
1:4 5 5
1:4 6 6
$END
$FRAG
1 1 1 1
S     1
SPZ 2
PX 2
PY 2
$END
$ORB
1 1 1 1 1 1
2
2
3
4
2
1
$END
```

For the second fragment, “1” in the first line of $FRAG means that the block contains basis functions located on one atom; “SPZ 2” means that the fragment includes the S and PZ basis functions in the second atom. The basis functions are described by groups of *s*, *p*, *d*, *f*, etc. For example, a fragment including S, PZ, DXX, DYY, DZZ, FZZZ, FXXZ, and FYYZ basis functions in atoms 1 and 2 should be described as

```
$FRAG
2
spzdxxyyzzfzzzxxzyyz 1 2
$END
```

Here “s” means basis function S, “pz” means basis function PZ, “dxxyyzz” means DXX, DYY and DZZ, and “fzzzxxzyyz” means FZZZ, FXXZ, and FYYZ. The ordering of basis functions are not compulsively defined, but the basis functions with the same type of *s*, *p*, *d* and *f* should be written together. For example, the above description can be written equivalently as

```
$FRAG
2
spzfzzzxxzyyzdxxyyzz 1 2
```

```
$END
```

or

```
$FRAG
2
spzfxxzyyzzzzdxxyyzz 1 2
$END
```

as users like.

NOTE:

If FRGTYP=ATOM without $FRAG, each atom is defined as one fragment. Thus, the $FRAG section in the first example is not essential and can be erased from the file.

## 4.6 ORB Section (Required when ORBTYP=HAO/GEN)

The first line describes the number of basis functions (or fragments) that are used for VB orbitals. For instance, *max*(*i*) means that the *i*-th orbital is expanded as *max*(*i*) functions (fragments), which are specified in the following lines. If the value of *max*(*i*) is 1, it means that the *i*-th orbital is not optimized. From the second line, the indices of basis functions are listed, where one orbital begins with one new line. Following is example:

```
4 4 2
3 4 5 6   ; orbital 1 is expanded with 4 basis functions (fragments)
4 3 5 6   ; orbital 2 is expanded with 4 basis functions (fragments)
1 2       ; orbital 3 is expanded with 2 basis functions (fragments)
```

NOTE:

*a.* It is important to emphasize again that the *n*-th VB basis function in $ORB section is NOT necessarily the *n*-th primitive basis function, but the *n*-th VB basis function specified in the BFI section.

*b.* It is suggested to write the most important basis function as the first one, as the program takes the first function as the “parent” function for the orbital if GUESS=UNIT. This can avoid potential problems in convergence.

*c.* If ORBTYP=OEO is chosen, the $ORB is not needed. All the orbitals will be delocalized in the whole system, which means orbitals will use all basis functions.

*d.* If the users want to freeze (not optimize) some orbitals in the calculation, simply assigning the number of basis functions (fragments) of the corresponding orbital to “0”. For example, “0*5 2 2” means that there are totally 7 VB orbitals and the first 5 will be frozen during SCF iterations. In this case, an initial guess should be provided either by “GUESS=READ” or “GUESS=MO”.

## 4.7 AIM Section

This section is relevant if OUTPUT=AIM is specified. The content of this section is an optional file name specified by users. This file name will be used as the WFN file name. By default, the content of WFN file will be stored in ".dat" file.

## 4.8 GUS Section

This section is required if GUESS=MO is specified. This section describes how VB orbital guess comes from MOs. An example of $GUS from $H_2$ calculation is shown below:

```
$GUS
1 1
2 1
$END
```

The example shows that both VB orbitals 1 and 2 will get the initial guess from MO 1. All orbitals should be specified in this section.

# 5. OUTPUT FILES

## 5.1 XMO File

The output of XMVB is stored in a file with extension “`xmo`”. The following is an example for module distribution:

```
         *************************************************************

         M   M    MM MM    M   M    MMMM        MMMM        MMM
          M M     M M M    M   M    M   M          M      M   M
           M      M M M     M M     MMMM   MMM  MMMM      M   M
          M M     M   M     M M     M   M       M         M   M
         M   M    M   M      M      MMMM        MMMM   M   MMM

       *************************************************************

                        XMVB 2.0-beta-20120607

       Cite this work as:

       (a) L. Song, Z. Chen, F. Ying, J. Song, X. Chen, P. Su, Y.
       Mo, Q. Zhang and W. Wu, XMVB 2.0: An Ab Initio Non-orthogo-
       nal Valence Bond Program, Xiamen University, Xiamen, 361005
       , China, 2012 (b) L. Song, Y. Mo, Q. Zhang, W. Wu, J.
       Comput. Chem. 2005, 26, 514.


    Job started at Thu Jun  7 20:46:27 2012
    Work    Directory    at    /export/home/fmying/PROGRAM_XMVB/gamess-xmvb-
git/vbtest/normdet_test   PID =       12148

    ---------------Input File---------------
   H2 HF METHOD
   $ctrl
   nstr=3 guess=read iscf=5 nao=2 nae=2 iprint=3
   orbtyp=hao
   $end
   $stru
   1 2
   1 1
   2 2
   $end
   $orb
   1 1
   1
   2
   $end
    ---------------End of Input--------------

    COPYING PRIMITIVE INTEGRAL TO XMVB INTEGRAL...


   Non-zero 2-e integral           =               55
   Total 2-e integral              =               55
   Non-zero percentage             =          100.00%

    Reading 2-e Integrals...
    Done
```

```
H2 HF METHOD

OPTIMIZATION METHOD: LBFGS WITH ANALYTICAL ORBITAL GRADIENT

Number of Structures:    3

The following structures are used in calculation:

  1 *****    1  2
  2 *****    1  1
  3 *****    2  2

Nuclear Repulsion Energy:       0.712409


--------------Initial Guess--------------
  2   2
0.6081806644   1   0.4886866644   2
0.6081806548   3   0.4886866745   4
--------------End of Guess--------------

VBDET is applied

                   4  Coefficients                    4  Independent

ITER    1 E =      -1.146268
GNORM =   0.000104 CHANGE IN ENERGY:    -1.1462680

ICOVG =                     1

Total Energy:      -1.14626805

First Excited:     -0.195977


The Last Change in Energy:   0.000000

Number of Iteration:    1

                ******  MATRIX  OF  OVERLAP  ******


           1          2          3
  1     1.000000   0.789807   0.789807
  2     0.789807   1.000000   0.453272
  3     0.789807   0.453272   1.000000



              ******  MATRIX  OF  HAMILTONIAN ******


           1          2          3
  1    -1.848145  -1.501074  -1.501074
  2    -1.501074  -1.495011  -0.998371
  3    -1.501074  -0.998371  -1.495011



             ******  COEFFICIENTS OF STRUCTURES ******

    1     0.79642  ******    1  2
    2     0.12679  ******    1  1
    3     0.12679  ******    2  2
```

```
           ******  COEFFICIENTS OF DETERMINANTS ******

                           a
                           b

  1     0.46714  ******    2
                           1
  2     0.46714  ******    1
                           2
  3     0.12679  ******    1
                           1
  4     0.12679  ******    2
                           2


           ******  WEIGHTS OF STRUCTURES ******

  1     0.79378  ******    1  2
  2     0.10311  ******    1  1
  3     0.10311  ******    2  2

      Lowdin Weights

  1     0.54871  ******    1  2
  2     0.22565  ******    1  1
  3     0.22565  ******    2  2

      Inverse Weights

  1     0.90330  ******    1  2
  2     0.04835  ******    1  1
  3     0.04835  ******    2  2


              ******  OPTIMIZED ORBITALS  ******


         1          2
1     0.608104   0.000000
2     0.488767   0.000000
3     0.000000   0.608104
4     0.000000   0.488767


    ******  ORBITALS IN PRIMITIVE BASIS FUNCTIONS ******


                   1          2
                 0.0000     0.0000
                   A          A
 1  H  1  S    0.608104   0.000000
 2  H  1  S    0.488767   0.000000
 3  H  2  S    0.000000   0.608104
 4  H  2  S    0.000000   0.488767


              ******  ORBITAL OVERLAP  ******


         1          2
1     1.000000   0.673255
2     0.673255   1.000000
```

```
              ******        DENSITY MATRIX         ******


                      1          2          3          4

   1  H  1  S    0.232264
   2  H  1  S    0.186684   0.150048
   3  H  2  S    0.204270   0.164183   0.232264
   4  H  2  S    0.164183   0.131963   0.186684   0.150048
ISCF = 5 currently does not support VB orbital densities


               ===============================================
                      XMVB ATOMIC POPULATION ANALYSIS
               ===============================================


                   ******  POPULATION AND CHARGE  ******

      ATOM          MULL.POP.    CHARGE          LOW.POP.     CHARGE
   1 H              1.000000    0.000000         1.000000    0.000000
   2 H              1.000000    0.000000         1.000000    0.000000

                   ******  ATOMIC SPIN POPULATION ******

       ATOM           MULL.POP.                     LOW.POP.

     1   H            0.000000                      0.000000
     2   H            0.000000                      0.000000


                          ******  BOND ORDER  ******

                ATOM 1        ATOM 2           DIST     BOND ORDER

                 1 H           2 H            0.743       0.951


                   ******    VALENCE ANALYSIS    ******

                                TOTAL       BONDED        FREE
              ATOM            VALENCE      VALENCE     VALENCE
            1 H                 1.000        0.951       0.049
            2 H                 1.000        0.951       0.049

                   ****** DIPOLE MOMENT ANALYSIS ******

               DX          DY          DZ         TOTAL

           0.000000    0.000000    0.000000    0.000000


Cpu for the Job:       0.07 (sec)

Job Finished at Thu Jun  7 20:46:27 2012
```

## 5.2 ORB File and GUS File

A file with extension “orb” is an output file of XMVB, which stores the optimized VB orbitals. The format is as follows:

*max*(1), *max*(2), …, *max*(*val*3)

# comment for orbital 1

*cvic*(1,1), *nvic*(1,1), *cvic*(1,1), *nvic*(2,1),…, *cvic*(*max*(1),1), *nvic*(*max*(1),1)

# comment for orbital 2

*cvic*(1,2), *nvic*(1,2), *cvic*(2,2), *nvic*(2,2),…, *cvic*(*max*(2),2), *nvic*(*max*(2),2)

…

# comment for orbital *n*

*cvic*(1,*val*3), *nvic*(1,*val*3), *cvic*(2,*val*3), *nvic*(2, *val*3),…, *cvic*(*max*(*val*3), *val*3), *nvic*(*max*(*val*3), *val*3)

where *max(i)* stands for the number of basis functions in *i*-th VB orbital, *nvic(j,i)* is the *j*-th basis function in *i*-th VB orbital and *cvic(j,i)* is the coefficient of *nvic(j,i)*. The lines starting with "#" are treated as comments.

For VBSCF and BOVB calculations, a file of orbital guess may be provided. For VBCI calculations, the guess from a previous VBSCF calculation is required with the extension "gus". The format of orbital guess file is exactly the same as ORB file. Initial guess files with or without comments are both supported by XMVB.

## 5.3 XDAT File

The file with extension "xdat" is an output file of XMVB. It keeps some other information such as the orbitals in original basis form. Using utility "moldendat" can read this file and put the VB orbitals to Gaussian and GAMESS output files and Gaussian fchk files. (See Chapter 7)

## 5.4 DEN File

If keyword "den" of $CTRL section, hybrid VB methods (DFVB, VBEFP, VBPCM and VBEFPPCM), or printing level larger than 1 are specified, one-electron density is saved to a file with "den" extension.

## 5.5 INFO File

This file stores the information of basis functions for the current system, including number of atoms, number of primitive basis functions, basis functions for each atom, and the type of each basis function. This file is essential for stand-alone distribution to carry out FRGTYP=ATOM/SAO and population analysis.

# 6. UTILITIES

## 6.1 Viewing VB Orbitals: Moldendat

Viewing VB orbitals is available. To do that, you need to run a utility, called “moldendat”:

```
moldendat MOfile vbdat [denfile] >&vbfile
```

where MOfile is an output file of Gaussian or GAMESS-US, or formatted Gaussian checkpoint file (.fchk); vbdat is a XMVB xdat file; if .fchk file is inputted, an optional XMVB density file with extension “.den” is also supported. The program will produce an “NEW” output file (vbfile) with the same format as input MO files, with which you can view VB orbitals with MOLDEN or MacMolPlt (for GAMESS-US only) packages.

## 6.2 Preparing Integrals: PREINT

This utility is developed to prepare integrals and MO orbitals for XMVB. To run PREINT, simply type the command as following:

```
preint input >&output
```

where input is the input file (see below) and output is the user-defined output file. A typical input file for $F_2$ molecule is shown below:

```
hf cc- pVTZ
0 1
F          0.000000  0.000000  0.000000
F          0.000000  0.000000  1.400000
```

The program provides three files: 1e.tem, containing 1-e integral and MO orbitals, 2e.tem, storing 2-e integrals, and the file “basinfo”(see section 5.5). These two files can be renamed to x1e.int and x2e.int and be used directly for stand-alone version.

The Basis sets supported by PREINT currently are: STO-3G, STO-6G, 3-21G, 6-31G, 6-31G*, 6-31+G*, 6-31G**, 6-311G, 6-311G*, 6-311+G*, cc-pVDZ, aug-cc-pVDZ, cc-pVTZ, and Lanl2DZ.

## 6.3 6D to 5D Integral Transformation: 6D25D

This utility transforms integrals from 6D type to 5D type. Currently the utility supports D-transformation only and not available for F-transformation.

To run the utility, typing the command as following:

```
6D25D MOfile
```

where MOfile is the output file of GAMESS-US or PREINT. The program will search the output file and find the D-type basis functions, and transforms the 6D-type integrals storing in 1e.tem and 2e.tem to 5D-type. The transformed 1-e and 2-e integrals are stored in files 1e.tem_5d and 2e.tem_5d respectively.

The utility currently can only transform the integrals. MO transformation from 6D- to 5D-type is not available.

# 7. TEST CALCULATIONS

## 7.1 VBSCF calculation of HF Molecule

**GAMESS-US INPUT FILE:**

```
$contrl runtyp=energy scftyp=rhf vbtyp=xmvb $end
$basis gbasis=n21 ngauss=3 $end
$data
HF molecule 3-21G basis set
C1
H   1.0   0.000000  0.000000  0.000000
F   9.0   0.000000  0.000000  0.900000
$end
```

**XMVB INPUT FILE:**

```
HF molecule, 3 structures
$ctrl
str=full nae=2 nao=2 # automatically generate all 3 structures
iprint=3 orbtyp=hao frgtyp=sao
$end
$frag
1 1 1 1
s 1
spz 2
px 2
py 2
$end
$orb
1 1 1 1 1 1
2
2
3
4
1
2
$end
```

NOTE:

1: VB structures are generated automatically by “STR=FULL NAO=2 NAE=2” so $STR is no needed.

2: VB orbitals are described with SAO. See the $FRAG and $ORB.

## 7.2 BOVB calculation of $F_2$ Molecule

**GAMESS-US INPUT FILE:**

```
$contrl runtyp=energy scftyp=rhf $end
$contrl vbtyp=xmvb $end
$basis gbasis=n31 ngauss=6 ndfunc=1 $end
$data
F2 molecule
C1
F   9.0   0.000000  0.000000 -0.724000
F   9.0   0.000000  0.000000  0.724000
 $end
```

**XMVB INPUT FILE:**

```
F2 molecule with BOVB
$ctrl
nstr=3 bovb guess=read
$end
$str
1:8 9 10
1:8 9 9
1:8 10 10
$end
$orb
8*4 3*4 8*2
 1  2  5  6  9 10 11 12
16 17 20 21 24-27
 1  2  5  6  9-12
16 17 20 21 24-27
 3  7 14
 4  8 15
18 22 29
19 23 30
 1  2  5  6  9-12
16 17 20 21 24-27
$end
```

NOTE:

1: A pre-proceeded VBSCF calculation has been done and the wave function is used as the initial guess in BOVB calculation. See keyword “GUESS=READ” in $CTRL. In this case, the initial guess must be given to XMVB, otherwise the program will exit with and error.

2: Structures and orbitals are described with abbreviations.

3: Orbitals are described with ORBTYP=GEN(default value), the ordering of the basis function is the same as in GAMESS-US.

## 7.3 VBSCF calculation of $CH_3Cl$ Molecule

**GAMESS-US INPUT FILE:**

```
$contrl runtyp=energy scftyp=rhf $end
$contrl vbtyp=xmvb $end
$system mwords=20 $end
$basis gbasis=n31 ngauss=6 ndfunc=1 $end
$data
CH3Cl
C1
Cl   17.0     0.000000    0.000000    0.000000
C     6.0     0.000000    0.000000    1.778500
H     1.0    -1.029975    0.000000    2.131541
H     1.0     0.514987    0.891984    2.131541
H     1.0     0.514987   -0.891984    2.131541
$end
```

**XMVB INPUT FILE:**

```
CH3Cl with 3 structures and hybrid orbitals
$ctrl
bprep nstr=3 iscf=5 nao=2 nae=2 boys
$end
$stru
1:6 7 8
1:6 7 7
1:6 8 8
$end
$orb
8*4 6 8 6 8
15-22
23-30
15-22
23-30
1-6
7-14
1-6
7-14
$end
$bfi
6 30
1 2 3 4 5 6
1 1 1 1 2 1
1 1 1 1 2 1 3 3
1 1 1
1 1 1 3 3
1 1 1
1 1 1 2 2
1.0 6
```

```
1.0 9
1.0 10
1.0 13
1.0 14 1.0 15
1.0 16

1.0 21
1.0 24
1.0 25
1.0 28
1.0 29 1.0 30
1.0 31
1.0 35 1.0 37 1.0 39
1.0 36 1.0 38 1.0 40

1.0 7
1.0 11
1.0 18

1.0 22
1.0 26
1.0 33
1.0 35 -0.5 37 -0.5 39
1.0 36 -0.5 38 -0.5 40

1.0 8
1.0 12
1.0 19

1.0 23
1.0 27
1.0 34
1.0 37 -1.0 39
1.0 38 -1.0 40
$end
```

NOTE:

1: BPREP is selected so that $BFI is needed. Core orbitals are frozen and valence basis functions are hybridized.

2: A Boys localization is requested. It is recommended to the users to perform the same VBSCF calculation without Boys localization as well and compare the difference between the orbitals of these two calculations.

## 7.4 $S_N2$ reaction with WFN output

**GAMESS-US INPUT FILE:**

```
$contrl runtyp=energy scftyp=rhf icharg=-1 mult=1 $end
$contrl vbtyp=xmvb $end
$system mwords=20 $end
$basis gbasis=n31 ngauss=6 ndfunc=1 $end
$data
SN2 Reaction
C1
F   9.0    0.000000   0.000000   0.000000
C   6.0    0.000000   0.000000   1.783800
H   1.0   -1.076500   0.000000   1.783800
H   1.0    0.538250   0.932276   1.783800
H   1.0    0.538250  -0.932276   1.783800
F   9.0    0.000000   0.000000   3.567600
$end
```

**XMVB INPUT FILE:**

```
SN2 reaction
$ctrl
str=full nao=3 nae=4
iscf=2 output=aim
den guess=read
bprep
$end
$orb
11*6 6 8 6 6 8 6
21-31
21-31
21-31
32-42
32-42
32-42
1-6
7-14
15-20
1-6
7-14
15-20
$end
$bfi
3 42
1 2 3
1 1 1 1 2 1
1 1 1 1 2 1 3 3
1 1 1 1 2 1
1 1 1
```

```
1 1 1 3 3
1 1 1
1 1 1
1 1 1 2 2
1 1 1
1.0 2
1.0 5
1.0 6
1.0 9
1.0 10 1.0 11
1.0 12

1.0 17
1.0 20
1.0 21
1.0 24
1.0 25 1.0 26
1.0 27
1.0 31 1.0 33 1.0 35
1.0 32 1.0 34 1.0 36

1.0 38
1.0 41
1.0 42
1.0 45
1.0 46 1.0 47
1.0 48

1.0 3
1.0 7
1.0 14

1.0 18
1.0 22
1.0 29
1.0 31 -0.5 33 -0.5 35
1.0 32 -0.5 34 -0.5 36

1.0 39
1.0 43
1.0 50

1.0 4
1.0 8
1.0 15

1.0 19
1.0 23
1.0 30
1.0 33 -1.0 35
```

```
1.0 34 -1.0 36

1.0 40
1.0 44
1.0 51
$end
$aim
sn2.wfn
$end
```

NOTE:

VBSCF calculation for the TS of $S_N2$ reaction is performed in this example. Note that WFN file output is requested by “OUTPUT=AIM” and the file name is “sn2.wfn”, as specified by $AIM.

## 7.5 $BH_3NH_3$ Complex

**GAMESS-US INPUT FILE:**

```
$contrl runtyp=energy scftyp=rhf $end
$contrl vbtyp=xmvb $end
$basis gbasis=n31 ngauss=6 ndfunc=1 $end
$data
BH3NH3
C1
N    7.0      0.000000   0.000000   0.000000
B    5.0      0.000000   0.000000   1.664112
H    1.0      0.951736   0.000000  -0.366785
H    1.0     -0.475868  -0.824228  -0.366785
H    1.0     -0.475868   0.824228  -0.366785
H    1.0     -1.171029   0.000000   1.967388
H    1.0      0.585515  -1.014141   1.967388
H    1.0      0.585515   1.014141   1.967388
 $end
```

**XMVB INPUT FILE:**

```
BH3NH3
$ctrl
nstr=2 itmax=500 orbtyp=hao
$end
$stru
1:8 9 10
1:8 9 9
$end
$orb
4*10
1 3 4 5
1 3 4 5
1 3 4 5
1 3 4 5
2 6 7 8
2 6 7 8
2 6 7 8
2 6 7 8
1 3 4 5
2 6 7 8
$end
```

NOTE:

VBSCF calculation for BH3NH3 is proceeded in this example. Note that the orbitals are described in terms of the atoms.

## 7.6 $CH_4$+$SiH_3$ H-abstract reaction

**GAMESS-US INPUT FILE:**

```
$contrl runtyp=energy scftyp=rohf mult=2 $end
$contrl vbtyp=xmvb $end
$basis gbasis=n31 ngauss=6 ndfunc=1 $end
$data
CH3--H--SiH3
C1
C    6.0     0.000000    0.000000    0.000000
H    1.0     0.000000    0.000000    1.445500
Si  14.0     0.000000    0.000000    3.166000
H    1.0     1.056497    0.000000   -0.253997
H    1.0    -0.528248   -0.914953   -0.253997
H    1.0    -0.528248    0.914953   -0.253997
H    1.0    -1.399179    0.000000    3.667976
H    1.0     0.699590    1.211725    3.667976
H    1.0     0.699590   -1.211725    3.667976
$end
```

**XMVB INPUT FILE:**

```
CH3--H--SiH3 BOVB
$ctrl
str=full nao=3 nae=3 nmul=2
bovb guess=read itmax=800
bprep
$end
$bfi
10 28
1 2 3 4 5 6 9 10 11 12
16 17
 2  5  6  9 10 11 12 37 38 39 40 41 42
23 26 27 30 31 32 33 43 44 45 46 47 48
$end
$orb
13 13 2 13 13
3-15
16-28
1 2
3-15
16-28
$end
```

NOTE:

TS of hydrogen abstract reaction is calculated with BOVB method. Core orbitals and π electrons are frozen in this case. An initial guess from a pre-proceeded VBSCF calculation should be provided.

## 7.7 BDO calculation of $C_2H_6$

**GAMESS-US INPUT FILE:**

```
$contrl runtyp=energy scftyp=rhf $end
$contrl vbtyp=xmvb $end
$basis gbasis=n31 ngauss=6 ndfunc=1 $end
$data
C2H6
C1
C      6.0      0.000000   0.000000   0.000000
C      6.0      0.000000   0.000000   1.531500
H      1.0      1.023907   0.000000  -0.396508
H      1.0     -0.511953  -0.886729  -0.396508
H      1.0     -0.511953   0.886729  -0.396508
H      1.0      1.008351   0.177800   1.928008
H      1.0     -0.658155   0.784358   1.928008
H      1.0     -0.350197  -0.962158   1.928008
$end
```

**XMVB INPUT FILE:**

```
c2h6
$ctrl
nstr=1 itmax=500 orbtyp= bdo guess=read
bprep
$end
$stru
1 2 3 4 5 6 7 8 9 10 11 12 13 14
$end
$orb
14 2 14 2 14 2 14 14 14 2 14 2 14 2
1-14
15-16
1-14
17-18
1-14
19-20
1-14
21-34
21-34
35-36
21-34
37-38
21-34
39-40
$end
$bfi
2 40
1 2
```

```
2 3 4 5 6 7 8 9 10 11 12 13 14 15 31 32 33 34 35 36 17 18 19 20 21 22 23 24 25 2
6 27 28
29 30 37 38 39 40 41 42
$end
```

NOTE:

1. A pre-proceeded VBSCF calculation without “BDO” is essential to provide good initial guess.
2. PPD algorithm is used in this case for faster speed by keyword VBFTYP=PPD.
3. The value of IPRINT should not be larger than 1, since PPD algorithm is not available for density calculation, which is needed for IPRINT=2 and IPRINT=3.

## 7.8 DFVB calculation of $H_2$+H reaction

**GAMESS-US INPUT FILE:**

```
$contrl runtyp=energy scftyp=rohf mult=2 $end
$contrl vbtyp=xmvb $end
$contrl dfttyp=lyp $end
$data
H--H--H
C1
H      1.0        0.000000    0.000000    0.000000
acct

H      1.0        0.000000    0.000000    0.931200
acct

H      1.0        0.000000    0.000000   -0.931200
acct

$end
```

**XMVB INPUT FILE:**

```
H--H--H abstraction
$ctrl
str=full nmul=2 nao=3 nae=3 dfvb den
orbtyp=hao
$end
$orb
1 1 1
1
2
3
$end
```

NOTE:

This example is a simple DFVB calculation. Structures are generated automatically and orbitals are expressed as terms of atoms.

# Appendix A. THEORY AND METHODOLOGY

In this appendix, a brief introduction to VB theory and methodologies will be given to the users. For more detailed information, it is recommended to the users to read our reviews [5,6] and research papers.[1-4,7,10-17]

## A.1 Introduction to VB Theory

In quantum chemistry, the many-electron wave function for a system is expressed as a linear combination of state functions:

$$\Psi = \sum_K C_K \Phi_K \tag{1}$$

In spin-free quantum chemistry, state functions $\Psi_K$ should be a spin eigenfunction with anti-symmetry with respect to permutation of electron indices.[1-3] The wave function is of the form

$$\Phi_K = \hat{A}\Omega_0\Theta_K \tag{2}$$

where $\hat{A}$ is an antisymmetrizer, $\Omega_0$ is an orbital product as

$$\Omega_0 = \phi_1(1)\phi_2(2)\cdots\phi_N(N) \tag{3}$$

where $\{\phi_i\}$ is the set of VB orbitals which can be purely localized hybrid atomic orbitals (HAOs), bond distorted orbitals (BDOs, delocalized along the bonding direction)[7], and totally delocalized overlap enhanced orbitals (OEOs), and $\Theta_K$ is a spin function. For VB methods, the state functions are VB functions, and their spin functions may be taken as the Rumer basis sets

$$\begin{aligned}\Theta_K &= 2^{-1/2}[\alpha(i_1)\beta(j_1)-\beta(i_1)\alpha(j_1)]\times 2^{-1/2}[\alpha(i_2)\beta(j_2)-\beta(i_2)\alpha(j_2)]\cdots \\ &= \prod_{(ij)} 2^{-1/2}[\alpha(i)\beta(j)-\beta(i)\alpha(j)]\prod_k \alpha(k),\end{aligned} \tag{4}$$

where $(ij)$ runs over all bonds and $k$ over all unpaired electrons. Given an orbital product $\Omega_0$, a complete set of VB functions is constructed by choosing all independent spin functions $\Theta_K$.

The coefficients $\{C_K\}$ in Eq. (1) are determined by solving the conventional secular equation **HC** = $E$**MC,** where Hamiltonian and overlap matrices are defined as follows:

$$H_{KL} = \langle \Phi_K | H | \Phi_L \rangle \tag{5}$$

and

$$M_{KL} = \langle \Phi_K | \Phi_L \rangle \tag{6}$$

Structural weights are given by the Coulson-Chirgwin formula

$$W_K = \sum_L C_K M_{KL} C_L \tag{7}$$

Eqs. (5) and (6) involve *N*! terms due to antisymmetrizer $\hat{A}$. If one-electron functions are orthogonal, only a few terms are non-zero and make contributions to the matrix elements, and consequently the matrix elements can be conveniently evaluated. However, in VB methods, non-orthogonal orbitals are generally used, and thus all *N*! terms make contributions to the matrix elements. Although it is not necessary to expand all *N*! terms to evaluate a determinant, the computational demanding in VB calculations is in general much more than that in MO calculations.

## A.2 The Evaluation of Hamiltonian and Overlap Matrices

In the XMVB package, two algorithms are implemented to compute the Hamiltonian and overlap matrices: one based on the Slater determinant expansion method, and the other based on the paired-permanent-determinant method.

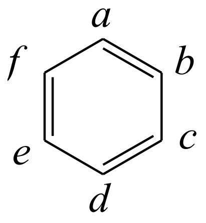
### I. Slater determinant expansion algorithm

Traditionally, a HLSP function is expressed in terms of $2^m$ Slater determinants (*m* is the number of covalent bonds of structure),

$$\Phi_K = \prod_i (1 - P_i) D(\Omega_K) \tag{8}$$

where $D(\Omega_K)$ is a Slater determinant corresponding to Eq. (3), $P_i$ is an operator that exchanges the spins of the two electrons forming the *i*-th bond.

**Example:** A HLSP function corresponding to a Kekulé structure of benzene



is written as

$$\begin{aligned}\Phi_K = & \left|a\bar{b}c\bar{d}e\bar{f}\right| - \left|\bar{a}bc\bar{d}e\bar{f}\right| - \left|a\bar{b}\bar{c}de\bar{f}\right| + \left|\bar{a}b\bar{c}de\bar{f}\right| \\ & - \left|a\bar{b}c\bar{d}\bar{e}f\right| + \left|\bar{a}bc\bar{d}\bar{e}f\right| + \left|a\bar{b}\bar{c}d\bar{e}f\right| - \left|\bar{a}b\bar{c}d\bar{e}f\right|\end{aligned}$$

The Hamiltonian matrix element is expressed as

$$\langle D_i | H | D_j \rangle = \sum_{r,s} f_{rs} D(S_r^s) + \sum_{r<u,s<t} (g_{rs,ut} - g_{rs,tu}) D(S_{ru}^{st}) \tag{9}$$

where $f_{rs}$ and $g_{rs,ut}$ are one-electron and two-electron integrals respectively, and $D(S_r^s)$ and $D(S_{ru}^{st})$ are the first and the second order cofactors of the overlap matrix between the two determinants respectively. Cofactors are computed by the Jacobi ratio theorem. The costs are of the order $N^3$ for the first order and $N^4$ for the second order cofactors at most.

### II. Paired-permanent-determinant approach[3]

Paired-permanent-determinant (PPD) approach is based on the spin-free form of VB theory. In the spin-free VB theory, the Hamiltonian and overlap matrix elements are now written as

$$H_{KL} = \langle \Phi_K | H | \Phi_L \rangle = \sum_{P \in S_N} D_{11}^{[\lambda]}(P) \langle \Omega_K | HP | \Omega_L \rangle \tag{10}$$

and

$$M_{KL} = \langle \Phi_K | \Phi_L \rangle = \sum_{P \in S_N} D_{11}^{[\lambda]}(P) \langle \Omega_K | P | \Omega_L \rangle \tag{11}$$

respectively, where is the first diagonal element of the standard irreducible representation of permutation $P$ of the symmetric group $S_N$. In the PPD approach, a function, called PPD, is defined as follow:

Given an $N \times N$ square matrix $\boldsymbol{A} = \{a_{ij}, i, j = 1, 2, \cdots, N\}$ the PPD of $\boldsymbol{A}$ for the irreducible representation $[\lambda]$ is the number

$$\mathrm{ppd}(\lambda, \mathbf{A}) = \sum_{P \in S_N} D_{11}^{[\lambda]}(P) a_{1p_1} a_{2p_2} \cdots a_{Np_N} \tag{12}$$

The evaluation of a PPD function is performed by a procedure similar to the Laplacian expansion algorithm for determinant. Hamiltonian and overlap matrix elements are computed by multiplying electronic integrals with their corresponding cofactors of PPDs. Evaluation of a PPD is more complicated than that of a determinant. But it can be beneficial when there are many bonded pairs in system. In that case there are only a few PPDs rather than numerous determinants to be evaluated.

## A.3 Orbital Optimization

The gradient vectors of energy are evaluated in four ways: the first is the numerical approximation by differential method; the second is analytical gradient based on Fock matrices, using only the first order density matrix; the third is analytical based on the first and the second order orbital density matrices; and the third is based on generalized Brillouin theorem. The first three methods are fitted for all-type orbitals, and the later one is only available for strictly localized and delocalized orbitals. The second one is suitable only when there is no orthogonality between VB functions.

There are two orbital optimization methods adopted in the package. The optimization with numerical gradient is based on the Davidson-Fletcher-Powell (DFP) family of variable metric methods, and the optimization with analytical gradient is proceeded with limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS)[8] method.

## A.4 The VBSCF Method

The wave function of Valence Bond Self Consistent Field (VBSCF)[9] method is the linear combination of VB structures, as shown in eq. (1). In VBSCF method, All VB structures share the same set of VB orbitals, and both sets of the structure coefficients and VB orbitals are optimized simultaneously to minimize the total energy. This is comparable to the MCSCF method in the MO theory. VBSCF method takes care of the static electron correlation and gives equivalent results to the MO-based CASSCF calculations. It should be noted that the dynamic electron correlation is not accounted for in the VBSCF method. In XMVB, VBSCF method is the default method, thus this keyword can be ignored.

## A.5 Post-VBSCF Methods

The VBSCF result includes only static correlation energy, which makes VBSCF results not accurate enough for quantitative researches. The purpose of post-VBSCF methods is to take dynamic correlation into account as much as possible to get accurate enough results. There are several post-VBSCF methods developed so far[10-13] and will be introduced in this section. It is strongly recommended to perform post-VBSCF calculations with initial guesses from a pre-proceeded VBSCF calculation. As to VBCI and VBPT2, this is enforced.

### A.5.1 The BOVB Method

The orbitals of Breathing Orbital Valence Bond (BOVB) [10] method are also optimized by SCF procedure, as VBSCF does. The difference between VBSCF and BOVB methods is that BOVB provides an extra degree of freedom during orbital optimization. In BOVB method, each VB structure has its own set of orbitals and are optimized independently.

$$\Psi^{\text{VBSCF}} = C_1\left(\left|\phi_a\overline{\phi_b}\right| - \left|\phi_b\overline{\phi_a}\right|\right) + C_2\left|\phi_a\overline{\phi_a}\right| + C_3\left|\phi_b\overline{\phi_b}\right| \tag{13a}$$

$$\Psi^{\text{BOVB}} = B_1\left(\left|\phi_a\overline{\phi_b}\right| - \left|\phi_b\overline{\phi_a}\right|\right) + B_2\left|\phi'_a\overline{\phi'_a}\right| + B_3\left|\phi''_b\overline{\phi''_b}\right| \tag{13b}$$

Thus, the orbitals adopt themselves to the instantaneous field of the VB structures, rather than to the mean field of all the structures in VBSCF. This degree of freedom makes the orbitals in BOVB “Breathing” in different structures, introduces dynamic correlation, and thereby improves considerably the accuracy of VB computations.

### A.5.2 The VBCI Method

The VBCI[11] method is based on localized VB orbitals. In this method VB orbitals are divided to several blocks (occupied and virtual orbitals). Excited VB structures are generated by replacing occupied VB orbitals with virtual orbitals that are localized on the same block. The wave function of VBCI is the linear combination of all reference and excited VB structures

$$\Psi^{\mathrm{VBCI}} = \sum_K \sum_i C_{Ki} \Phi_K^i \tag{14}$$

where $\Phi_K^i$ is CI structure coming from VBSCF structure $K$, including reference and excited structures, and the coefficients $\{C_{Ki}\}$ are obtained by solving the secular equation.

The VBCI weight can be given either with eq. (7), which gives weights of all CI structures, or in a more compact way as

$$W_K = \sum_i W_{Ki} \tag{15}$$

where $W_K$ is the contracted weights of reference structure $K$, including weights of all CI structures coming from structure $K$.

Allowing for different excitations for different electronic shells, currently the VBCI method consists of the following calculation levels:

**VBCI(S,S):** only single excitations are involved in either active electron or inactive electron. In brief, this is a VBCIS procedure.

**VBCI(D,S):** the active shell is treated by single and double excitations, whereas the inactive shell by single excitations only. Also included in this level are double excitations which consist of a single excitation from each shell.

**VBCI(D,D):** single and double excitations are involved for both active and inactive electrons, in short, VBCISD.

### A.5.3 The VBPT2 Method

Another post-VBSCF method is Valence Bond second-order Perturbation Theory (VBPT2)[12] method. The wave function of VBPT2 can be separated into 2 parts as

$$\Psi^{\mathrm{VBPT2}} = \Psi^0 + \Psi^1 \tag{16}$$

where VBSCF wave function is taken as the zeroth-order wave function $\Psi^0$, and the first-order part is the combination of singly and doubly excited wave functions

$$\Psi^1 = \sum_{R \in V^{SD}} C_R^1 \Phi_R \tag{17}$$

To enhance the efficiency of VBPT2, the virtual orbitals are delocalized and orthogonal to the occupied space, and the excitations include all virtual orbitals. In this manner, the excited structures in VBPT2 don't belong to any fundamental structure, and the matrix elements can be calculated easily with Coulson-Slater rules.

### A.5.4 The DFVB Method

Density functional valence bond(DFVB)[13] method is a VB computational method which combines VBSCF and DFT correlation functional. In DFVB method, the wave function, density and static correlation energy is provided by VBSCF method, while the dynamic correlation energy is obtained by DFT correlation functional. The total energy formalism of DFVB is expressed as:

$$E^{\mathrm{DFVB}} \approx E^{\mathrm{VBSCF}} + E_{\mathrm{C}}\left[\rho^{\mathrm{VB}}\right] \tag{18}$$

where $E^{\mathrm{VBSCF}}$ is the VBSCF part, including static correlation energy, and $E_{\mathrm{C}}\left[\rho^{\mathrm{VB}}\right]$ is the DFT correlation energy obtained by a functional according to current VB density $\rho^{\mathrm{VB}}$. The total energy and wave function is optimized through a SCF procedure. In practice, GGA correlation functionals, such as LYP, PW, PBE correlation functional, are recommended functionals.

## A.6 Solvation VB Methods

### A.6.1 The VBPCM Method

The VBPCM[14] method is an *ab initio* solvation VB method that is based on implicit solvation model PCM in which the state wave function is expressed in the usual terms as a linear combination of VB structures. The Schrödinger equation of VBPCM is expressed as

$$\left(\hat{H}^0 + \hat{V}^{\mathrm{PCM}}\right)\Psi^{\mathrm{VBPCM}} = E^{\mathrm{VBPCM}}\Psi^{\mathrm{VBPCM}} \tag{19}$$

where $\hat{H}^0$ is the Hamiltonian operator in vacuum and $\hat{V}^{\mathrm{PCM}}$ is the solvation potential obtained by PCM.[18] The VBPCM wave function and energy are optimized simultaneously in an SCF procedure. VBPCM is now available for VBSCF and BOVB.

VBPCM has been used in several researches.[15,16] VBPCM has been rewritten in XMVB 2.0 and is capable for hetero-PCM and EFP/PCM calculations.

### A.6.2 The VBEFP Method

The VBEFP[17] method is an QM/MM method in which the QM part is expressed as a VB wave function and the MM part is expressed with EFP1,[19,20] which is a polarized water model proposed by Gordon et al. The VBEFP energy is obtained by following equation

$$\left(\hat{H}^0 + \hat{V}^{\mathrm{EFP}}\right)\Psi^{\mathrm{VBEFP}} = E^{\mathrm{VBEFP}}\Psi^{\mathrm{VBEFP}} \tag{20}$$

where $\hat{H}^0$ is the Hamiltonian operator in vacuum and $\hat{V}^{\mathrm{EFP}}$ is the solvation potential obtained by EFP method. An SCF procedure is used to optimize the wave function and energy of VBEFP simultaneously.

Currently, VBEFP is available only in module distribution and is only available for VBSCF. With the use of EFP, VBEFP is very useful to take strong and short solvent-solute interactions into account.

# Appendix B. REFERENCES

[1] Zhang, Q.; Li, X. *J. Mol. Struct.* 1989, 198, 417.

[2] Wu, W.; Mo, Y,; Zhang, Q.; *J. Mol. Struct.* (Theochem), 1993, 283, 227

[3] (a) Wu, W.; Wu, A.; Mo, Y.; Zhang, Q. *Science in China* 1996, B39, 456. (b) Wu, W.; Wu, A.; Mo, Y.; Lin, M.; Zhang, Q. *Int. J. Quant. Chem.* 1998, 67, 287. (c) Song, L.; Luo, Y.; Dong, K.; Wu, W.; Mo, Y.; Zhang, Q. *Science in China* 2001, B44, 561. (d) Wu, W.; Mo, Y.; Cao, Z.; Zhang, Q. Valence Bond Theory, D. L. Cooper ed., Elsevier Science, Amsterdam., p143, 2002.

[4] Song, L.; Mo, Y.; Zhang Q.; Wu, W. *J. Comput. Chem.* 2005, 26, 514.

[5] Su, P.; Wu, W. *WIREs Comput. Mol. Sci.* 2012, ASAP.

[6] Wu, W.; Su, P.; Shaik, S.; Hiberty, P. C. *Chem. Rev.* 2011, 111, 7557.

[7] Mo, Y.; Lin, Z.; Wu, W.; Zhang, Q. *J. Phys. Chem.* 1996, 100, 11569.

[8] Nocedal, G.; *Math. Comput.* 1980, 35, 773

[9] (a) Verbeek, J.; van Lenthe. J. H. *J. Mol. Struct.* (Theochem) 1991, 229, 115. (b) van Lenthe, J. H. *Int. J. Quant. Chem.* 1991, 40, 201. (c) Balint-Kurti, G. G.; Benneyworth, P. R.; Davis, M. J.; Williams, I. H. *J. Phys. Chem.* 1992, 96, 4346.

[10] (a) Hiberty, P. C.; Flament, J. P.; Noizet, E. *Chem. Phys. Lett.* 1992, 189, 259. (b) Hiberty, P. C.; Humbel, S.; Byrman, C. P.; Van Lenthe, J. H. *J. Chem. Phys.* 1994, 101, 5969. (c) Hiberty, P. C.; Humbel, S.; Archirel, P. *J. Phys. Chem.* 1994, 98, 11697.

[11] (a) Wu, W.; Song, L.; Cao, Z.; Zhang, Q.; Shaik, S. *J. Phys. Chem. A* 2002, 105, 2721. (b) Song, L; Wu, W.; Zhang, Q.; Shaik, S. *J. Comput. Chem.* 2004, 25, 472.

[12] Chen, Z.; Song, J.; Shaik, S.; Hiberty, P. C.; Wu, W. *J. Phys. Chem. A* 2009, 113, 11560.

[13] Ying, F.; Su, P.; Che, Z.; Shaik, S.; Wu, W. *J. Chem. Theor. Comput.* 2007, 8, 1608

[14] Song, L.; Wu, W.; Zhang, Q.; Shaik, S. *J. Phys. Chem. A* 2004,108, 6017.

[15] Su, P.; Ying, F.; Wu, W.; Hiberty, P. C.; Shaik, S. *ChemPhysChem* 2007, 8, 2603.

[16] Su, P.; Song, L.; Wu, W.; Shaik, S.; Hiberty, P. C. *J. Phys. Chem. A* 2008, 112, 2088.

[17] Ying, F.; Chang, X.; Su, P.; Wu, W. *J. Phys. Chem. A* 2012, 116, 1846.

[18] Tomasi, J.; Mennuci, B.; Cammi, R. *Chem. Rev.* 2005, 105, 2999.

[19] Day, P.; Jensen, J. H.; Gordon, M. S.; Webb, S. P.; Stevens, W. J.; Krauss, M.; Garmer, D.; Basch, H.; Cohen, D. *J. Chem. Phys.* 1996, 105, 1968.

[20] Gordon, M. S.; Freitag, M. A.; Bandyopadhyay, P.; Jensen, J. H.; Kairys, V.; Stevens, W. J. *J. Phys. Chem. A* 2000, 105, 293.