A Practical Guide for the XMVB Program

Tutors :

- Tristan Ribeyre
- Benoît Braïda

Outline

- General introduction
- How to write input files?
- How to get information from output files?

General introduction

An Ab Initio Non-orthogonal Valence Bond Program

Version 2.0 (2012)

Lingchun Song, Zhenhua Chen, Fuming Ying, Jinshuai Song, Xun Chen,

Peifeng Su, Yirong Mo, Qianer Zhang, Wei Wu*

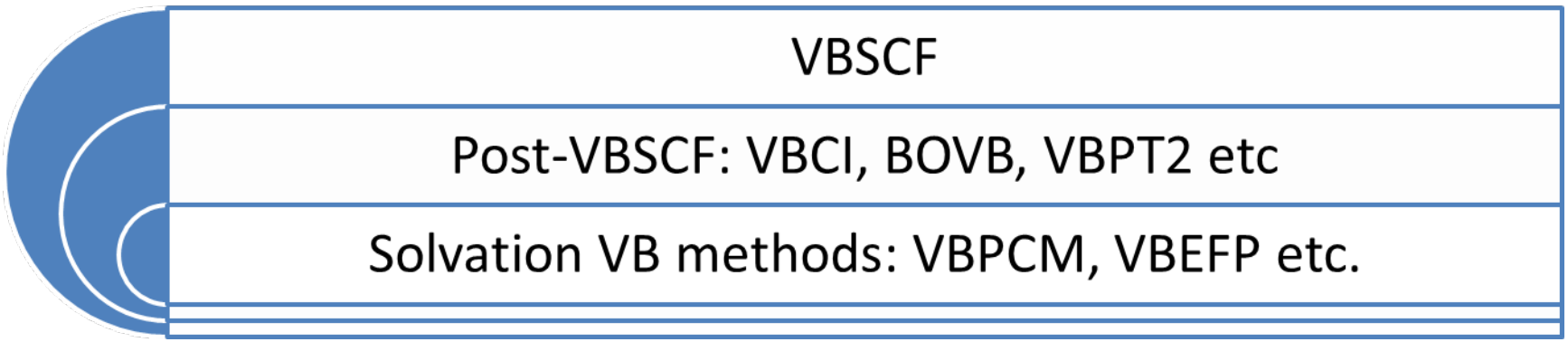
State Key Laboratory of Physical Chemistry of Solid Surfaces,

Fujian Provincial Key Laboratory of Theoretical and Computational Chemistry,

and Department of Chemistry,

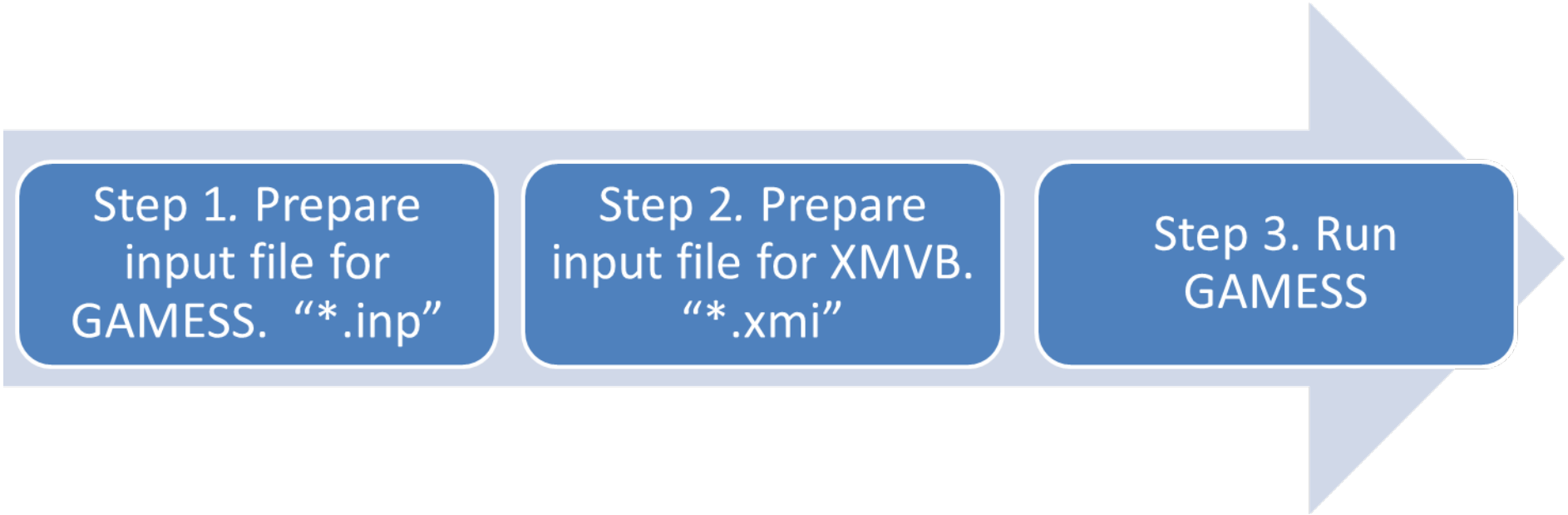
Xiamen University, Xiamen, Fujian 361005, CHINA

Current capabilities of XMVB 2.0



- Energy (Total, individual), Weights of structure
- Resonance energy
- Properties: bond order, atomic charges, dipole etc

How to Run An XMVB Job?



For a "XXX.xmi" file, you must prepare a corresponding "XXX.inp" file
The job name "XXX" for the two input files should be the same.

GAMESS

XMVB module

On the local cluster (Obelix)

Add my /bin to your $path :

export PATH= /users/lct/braida/bin:$PATH

Add it also preferably into your .bash_profile file

check : which vbrun

GAMESS

XMVB module

On the local cluster (Obelix)

1) Prepare : job.inp job.xmi

↓

2) run :

vbrun job

Batch script Name of your job **without extension**

↓

job.out

job .xmo, job.xdat, job.orb

job.sub.o2xxx (produced by the Batch system)

Useful documents

GAMESS manual

http://www.msg.ameslab.gov/gamess/documentation.html

XMVB manual

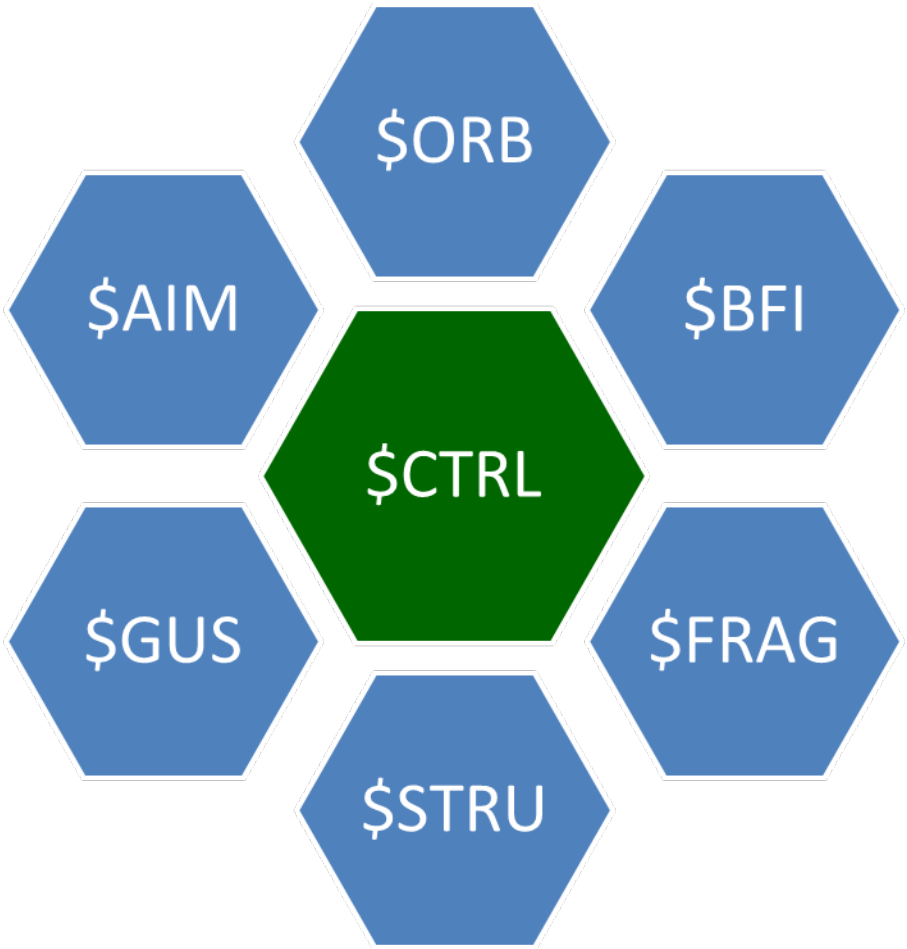
https://wiki.lct.jussieu.fr/workshop/images/7/71/XMVB_Manual_V20.pdf

Tutorial exercises

https://wiki.lct.jussieu.fr/workshop/index.php/VB_tutorial

How to write input files?

Structure of Typical XMVB Input File



Optional

Necessary

Example 1. H_2 molecule

h2.inp / **GAMESS INPUT FILE**

```
$contrl runtyp=energy scftyp=rhf $end
$contrl vbtyp=xmvb $end
$basis gbasis=n31 ngauss=6 NPFUNC=1 $end
$data
H2 molecule
C1
H   1.0   0.000000  0.000000  0.000000
H   1.0   0.000000  0.000000  0.742000
$end
```

→ rhf for mult=1
rohf or uhf for mult >1

The detailed information of GAMESS input file is available in GAMESS manual.

GAMESS input files (job.inp files) **will be provided for all exercises**

h2.xmi / **XMVB INPUT FILE**

```
$ctrl
nstr=3  iscf=5 nao=2 nae=2 boys
orbtyp=hao frgtyp=sao
$end
$str
1 2
1 1
2 2
$end
$frag
1 1
spz 1
spz 2
$end
$orb
1 1
1
2
$end
```

Options for the VB calculation

Define the VB structures

Define blocks of basis functions

Define orbitals

h2.xmi / **XMVB INPUT FILE**

```
$ctrl
nstr=3  iscf=5 nao=2 nae=2 boys
orbtyp=hao frgtyp=sao
$end
$str
1 2
1 1
2 2
$end
$frag
1 1
spz 1
spz 2
$end
$orb
1 1
1
2
$end
```

iscf : orbital optimization algorithm
nao: number of active orbitals
nae: number of active electrons

Best options :
- VBSCF : iscf=5
- BOVB : iscf=2
- VBCI : iscf=1

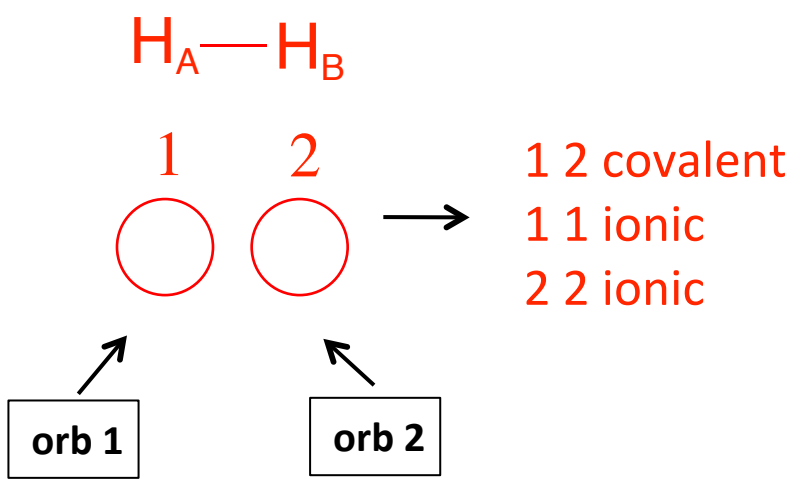
h2.xmi / **XMVB INPUT FILE**

```
$ctrl
nstr=3  iscf=5 nao=2 nae=2 boys
orbtyp=hao frgtyp=sao
$end
$str
1 2
1 1
2 2
$end
$frag
1 1
spz 1
spz 2
$end
$orb
1 1
1
2
$end
```

nstr =3, the number of structures
$str : describing VB structure or determinant.
In $str, a typical structure is written as:

a a b b c c … m n

a b c : double occupied orbitals
m n : single occupied orbitals



h2.xmi / **XMVB INPUT FILE**

```
$ctrl
nstr=3  iscf=5 nao=2 nae=2 boys
orbtyp=hao frgtyp=sao
$end
$str
1 2 ; covalent
1 1 ; ionics
2 2
$end
$frag
1 1
spz 1
spz 2
$end
$orb
1 1
1
2
$end
```

← It is possible to add free comments after «;»

h2.xmi / **XMVB INPUT FILE**

```
$ctrl
str=full  iscf=5 nao=2 nae=2 boys
orbtyp=hao frgtyp=sao
$end
```

Automatically generate structures based on the active orbitals and electrons.
str = full => generate all VB structures
cov => only covalent

nao = m, the final m orbitals are selected as active orbitals.
nae = n, the number of active VB electrons

Using str, $str can be absent.

```
$frag
1 1
spz 1
spz 2
$end
$orb
1 1
1
2
$end
```

h2.xmi / **XMVB INPUT FILE**

```
$ctrl
str=full iscf=5 nao=2 nae=2 boys
orbtyp=hao frgtyp=sao
$end
$str




$frag
1 1
spz 1
spz 2
$end
$orb
1 1
1
2
$end
```

orbtyp=hao, hybrid atomic orbitals. It is localized non-orthogonal orbitals

h2.xmi / **XMVB INPUT FILE**

```
$ctrl
str=full iscf=5 nao=2 nae=2 boys
orbtyp=hao frgtyp=sao
$end
```

frgtyp=sao, fragments are defined with **s**ymmetrized **a**tomic **o**rbitals.

Fragments are a series of basis function sets.

```
$frag
1 1
spz 1
spz 2
$end
$orb
1 1
1
2
$end
```

H_A H_B

MO				Basis functions
			1	
1	H	1	S 0.317976	1 H 1 S → frag 1
2	H	1	S 0.275854	2 H 1 S → frag 1
3	H	1	X 0.000000	3 H 1 X
4	H	1	Y 0.000000	4 H 1 Y
5	H	1	Z 0.018271	5 H 1 Z → frag 1
6	H	2	S 0.317976	6 H 2 S → frag 2
7	H	2	S 0.275854	7 H 2 S → frag 2
8	H	2	X 0.000000	8 H 2 X
9	H	2	Y 0.000000	9 H 2 Y
10	H	2	Z -0.018271	10 H 2 Z → frag 2

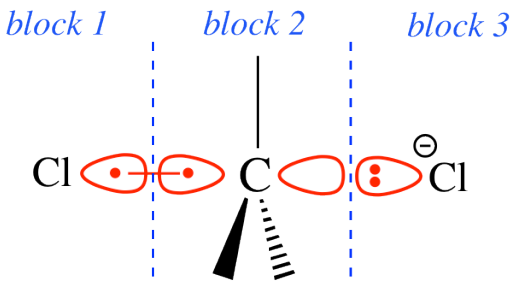
$frag section defines blocks of basis functions

chosen depending on localization (user choice) and symmetry properties

Fragments are a series of basis function sets.

A MORE ADVANCED EXAMPLE

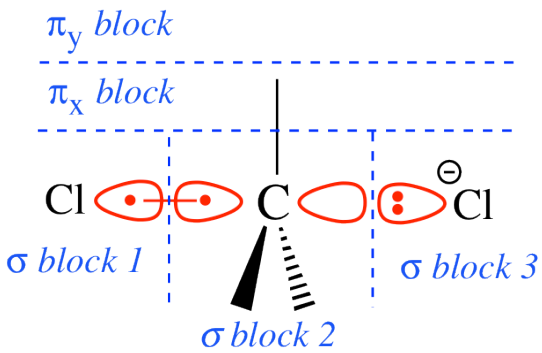
- A possible choice :



⇒ 3 fragments, and

3 blocks defined in $frag

- Another choice :



⇒ 3 fragments, but

5 blocks defined in $frag

$frag section defines blocks of basis functions

chosen depending on localization (user choice) and symmetry properties

h2.xmi / **XMVB INPUT FILE**

```
$ctrl
str=full iscf=5 nao=2 nae=2 boys
orbtyp=hao frgtyp=sao
$end
```

```
$frag
1 1
spz 1
spz 2
$end
```

$frag: description for fragments
Line 1: the number of atoms in each fragment.
Line 2 : the basis functions of s and pz in atom 1 are selected for the first basis bloc
Line 3 : the basis functions of s and pz in atom 2 are selected for the second basis bloc

```
$orb
1 1
1
2
$end
```

$orb: description for VB orbitals
Line 1: the number of fragments in each orbital.
Line 2: orbital 1 is described by fragment 1
Line 3: orbital 2 is described by fragment 2.

Summary (1)

1. The assignment of orbital optimization algorithm (iscf)
2. The writing of VB structures with $str (optional)
3. The determination of VB orbitals with ‘frgtyp=sao’

Example 2. L-BOVB calculation for HF

Using VBSCF orbitals as guess orbitals

cp hf.orb hf-bovb.gus

Preparing GAMESS and XMVB input files for L-BOVB

hf-bovb.inp

hf-bovb.xmi

Running XMVB

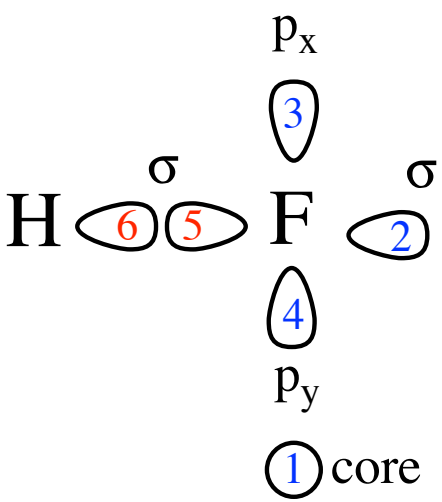
hf-bovb.xmi

HF molecule, 3 structures
$ctrl
nstr=3 nae=2 nao=2 bovb
iprint=3 orbtyp=hao frgtyp=sao iscf=2
guess=read
$end
$str
1 1 2 2 3 3 4 4 5 6
1 1 2 2 3 3 4 4 5 5
$end
…
…
…

hf-bovb.xmi

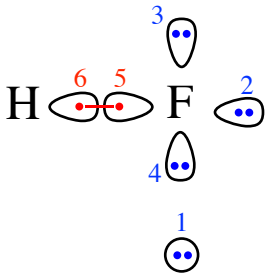
HF molecule, 3 structures
$ctrl
nstr=3 nae=2 nao=2 bovb
iprint=3 orbtyp=hao frgtyp=sao iscf=2
guess=read
$end
$str
1 1 2 2 3 3 4 4 5 6
1 1 2 2 3 3 4 4 5 5
$end
…
…
…

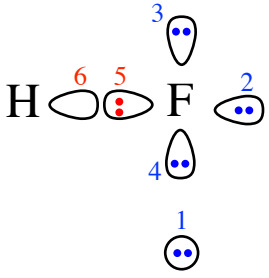
- In each structure, the occupation of the different orbitals is defined.
- The electrons are spin coupled **by pairs**



hf-bovb.xmi

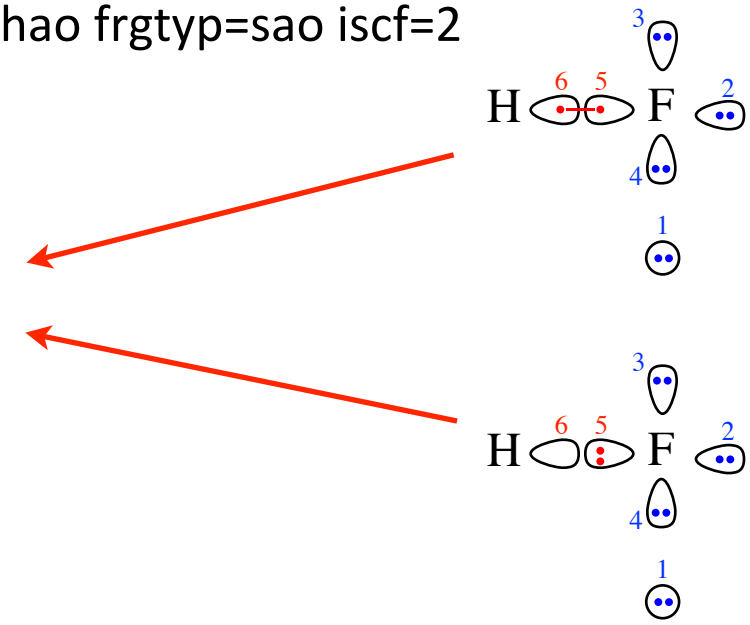
HF molecule, 3 structures
$ctrl
nstr=3 nae=2 nao=2 bovb
iprint=3 orbtyp=hao frgtyp=sao iscf=2
guess=read
$end
$str
1 1 2 2 3 3 4 4 5 6
1 1 2 2 3 3 4 4 5 5
$end
…
…
…





hf-bovb.xmi

HF molecule, 3 structures
$ctrl
nstr=3 nae=2 nao=2 bovb
iprint=3 orbtyp=hao frgtyp=sao iscf=2
guess=read
$end
$str
1:4 5 6
1:4 5 5
$end
…
…
…



Simplified input for structures

Before starting your first BOVB calculation, please have a look to :

https://wiki.lct.jussieu.fr/workshop/index.php/General_guidelines_for_BOVB_calculations

Example 3. VBCI calculation for HF

a VBSCF calculation with boys localization.

cp hf.orb hf-vbci.gus

Preparing GAMESS and XMVB input files for VBCI

hf-vbci.xmi
hf-vbci.inp

Running XMVB.

hf-vbci.xmi

hf.xmi

F2 molecule
$ctrl
nstr=3 boys
$end
…
…
…

HF molecule
$ctrl
str=full nae=2 nao=2 vbcisd
iprint=3 orbtyp=hao frgtyp=sao guess=read
$end
…
…
…

Example 4. D-BOVB calculation for F_2

Using L-BOVB orbital as guess orbitals

cp f2-l-bovb.orb f2-d-bovb.gus

Preparing GAMESS and XMVB input files for D-BOVB

f2-d-bovb.inp

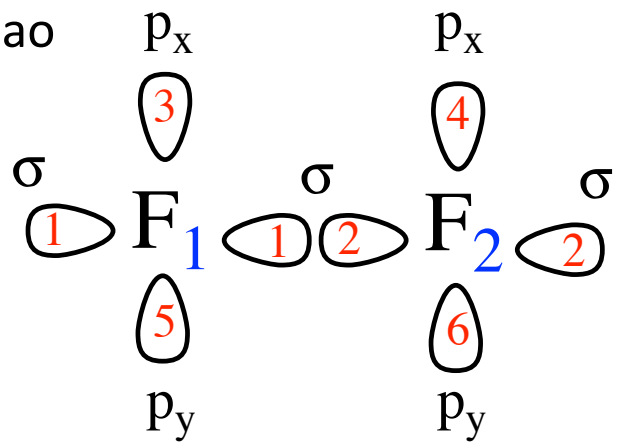
f2-d-bovb.xmi

Running XMVB

f2-l-bovb.xmi

F2 molecule, 3 structures, initial L-BOVB calculation

```
$ctrl
str=full iscf=2 nao=2 nae=2 orbtyp=hao frgtyp=sao
guess=read bovb
$end
$frag
1 1 1 1 1 1
spzdxxyyzz 1 ; block 1
spzdxxyyzz 2 ; block 2
pxdxz 1      ; block 3
pxdxz 2      ; block 4
pydyz 1      ; block 5
pydyz 2      ; block 6
$end
…
…
…
```



2 fragments (containing 1 atoms here), but
6 orbital blocks taking symmetry into account

The numbering of atoms corresponds to the one in the GAMESS job.inp input

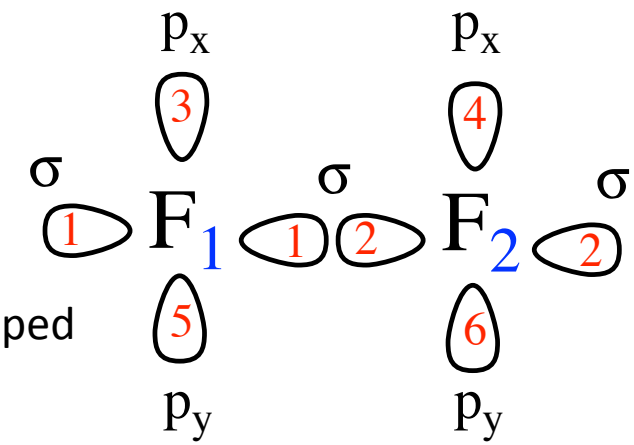
f2-l-bovb.xmi

F2 molecule, 3 structures, initial L-BOVB calculation

```
$ctrl
...
...
...
$orb
1*10
1  ; inactive
2
1
2
3
4
5
6
1 ; active
2
$end
```

10 orbitals (2 cores, 6 valence lone pairs, 2 actives) each developed on the basis functions of **1** block

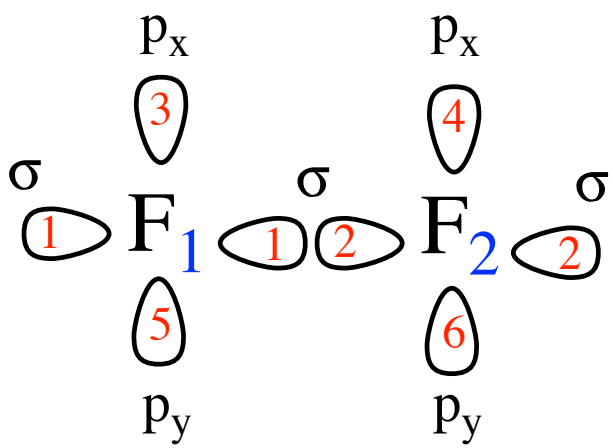
10 lines, where the number on each line refers to the block of basis functions on which the orbital is developed



f2-d-bovb.xmi

F2 molecule, 3 structures, D-BOVB calculation

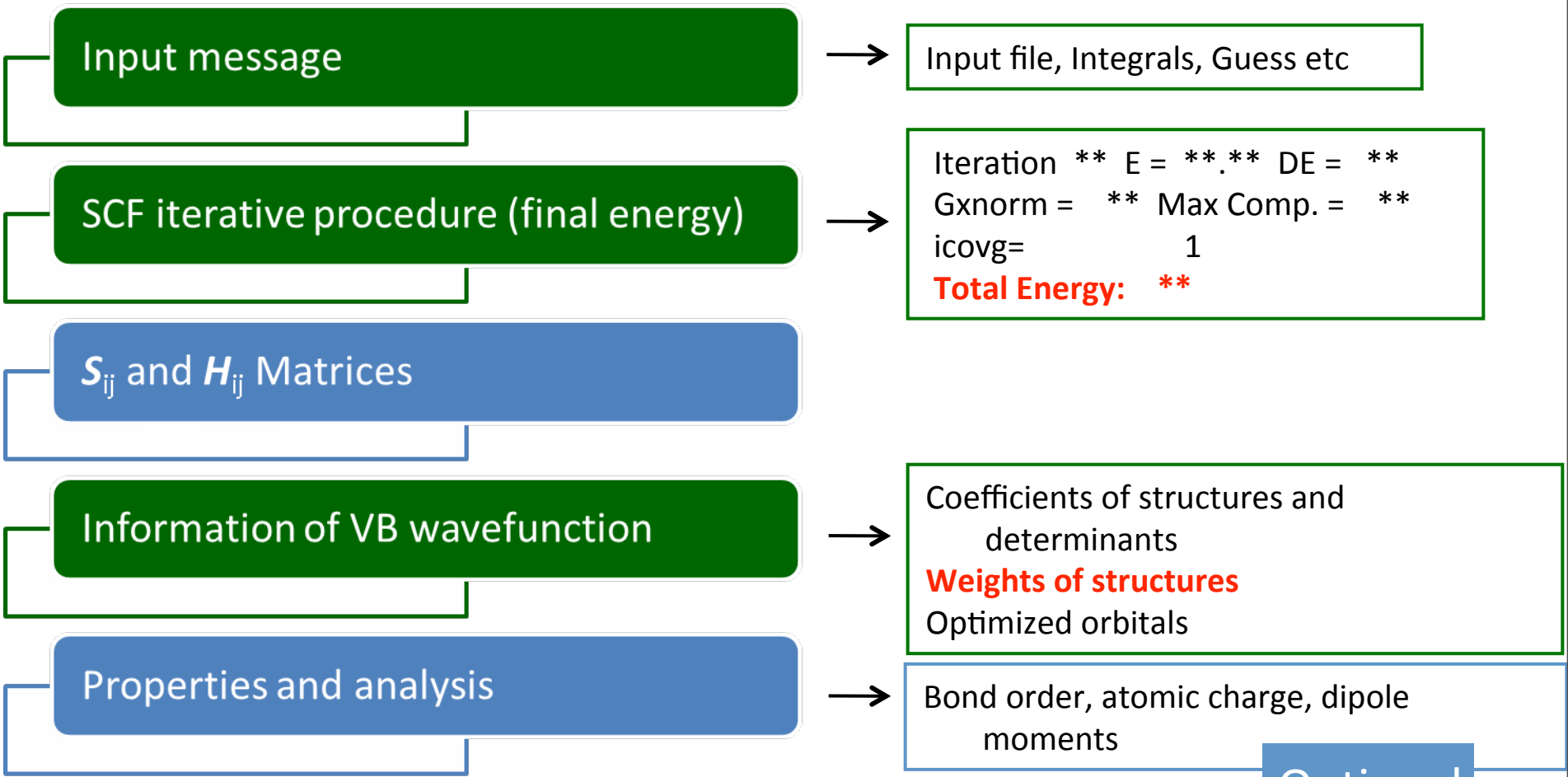
```
$ctrl
...
...
...
$orb
1*4 2*4 1*2
1  ; inactive
2
1
2
3 4
4 3
5 6
6 5
1 ; active
2
$end
```



Inactive π orbitals are now delocalized
both sigma active and inactive remains localized

How to get information from output files?

Information of XMO file



The VB calculation message in gamess output file is not correct

```
ITER   20 E =    -100.002122
ICOVG =                  1
Boys localizing
...
Total Energy:    -100.00212152
```

SCF procedures (final energy)

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

          1          2          3
 1    1.000000   0.630143   0.630143
 2    0.630143   1.000000   0.247723
 3    0.630143   0.247723   1.000000

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

          1          2          3
 1   -2.370630  -1.835188  -1.509357
 2   -1.835188  -2.248158  -0.822557
 3   -1.509357  -0.822557  -1.314986
```

S_{ij} and H_{ij} matrices

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

   1    0.56496  ******  1:4   5  6
   2    0.42059  ******  1:4   5  5
   3    0.01445  ******  1:4   6  6
```

Weights of VB structures

It’s up to you !