Changes in VB2000 from version 2.7 to 2.8

Each change will be introduced by a paragraph in black print. This may be followed by a paragraph in red print that outlines any changes needed in the GAMESS(US) code, a paragraph in purple print that indicates the new test input files for the new features, and a paragraph in blue print that discusses the changes to the VB2000 code in more detail. The file “Changelog” in VB2000/SRC contains changes up to version 2.7 and mostly to version 2.8 from the first time the code was submitted to the GAMESS group

VB to follow methods other than HF in GAMESS.

In version 2.7, the GAMESS run is a single point at RHF or ROHF. This is followed by a VB calculation at that geometry using the molecular orbitals to construct LMOs that may be used for the initial guess. These HF methods have been extended to use UHF or DFT, with the α orbitals used for the LMOs in UHF. Post HF methods can also be used, but the HF orbitals are used for the LMOs. This allows a quick comparison of the VB energy with say MP2, CCSD, etc. Similarly various MCSCF methods can be used. The HF core and MCSCF active orbitals are used to get the LMOs. However, a new directive $READMOS gets the LMOS from the HF orbitals if they are available in the $VEC group for a MOREAD set of orbitals as is normal for MCSCF.

This mainly involves removing code in inputa.src that tested for only HF runs. Some changes are necessary to allow XMVB to work properly and I hope I have fixed the code to work correctly with XMVB. The XMVB developers explain changes to be made to inputa.src in their documentation.

Test data exam-vb26 differs from exam-vb22 by using UHF rather than ROHF.
Test data exam-vb27 uses DFT B3LYP in GAMESS.
Test data exam-vb28 is like exam-vb27 but uses RHF/
Test data exam-vb29 tests the use of MCSCF starting orbitals from a CASSCF(6,6) calculation on benzene, followed by a SC(6) calculation which can be directly compared with the CASSCF.
Test data exam-vb31 is like exam-vb28, but does an MP2 evaluation in GAMESS,
Test data exam-vb32 is like exam-vb28, but does an CCSD(T) evaluation in GAMESS,
Test data exam-vb33 is like exam-vb29 but uses GUGA rather than ALDET for the MCSCF.

Subroutine GETVEC in GMS version implements the new directive $READMOS to read MOs from a $VEC group

VB to follow geometry optimisation or Hessian calculation in GAMESS.

Similar to the change outlined in the section above, it is now possible to run a geometry optimisation or Hessian calculation in GAMESS. This is followed by a VB calculation at that geometry using those molecular orbitals to construct the LMOs. The directive $NOVBPROP is used to indicate that the optimisation is not for the VB function but for the initial GAMESS RHF or whatever.

Minor changes are made to statpt.src and hess.src which include a new entry point to VBGMS in vb2gms.src. Note that vb2gms.src no longer now has to inspect the GAMESS data by calling NAMEIO. This means that it does not have to be updated to reflect updates in GAMESS for new options.

Test data exam-vb34 uses $NOVBPROP to optimise the geometry at the RHF level, followed by a VB calculation.
Test data exam-vb35 uses $NOVBPROP to do a VB calculation after a GAMESS Hessian run.

The new entry point in vb2gms.src is part of a major rewrite that simplifies and updates the code.

GVB in GAMESS followed by VB.

A GVB run in GAMESS can now precede a VB2000 run. The GVB orbitals in the CI form are transformed in VB2000 to the VB-like form as the initial guess. This is not as successful as you might expect, but it is useful in some cases. It is not available for open-shell cases.

No changes are needed in the GAMESS code.

Test data exam-vb30 tests the GVB start.
Test data exam-vb36 is a more realistic GVB start.

A new subroutine GETGVB in VB2GMS transforms the GVB orbitals from the CI to VB form and subroutine MODGVB reorders the GVB orbitals.

Generalized Product Function Energy Partitioning - GPF-EP

The Generalized Product Function Energy Partitioning (GPF-EP) is a powerful Energy Decomposition Analysis (EDA) scheme suited for studying the nature of the chemical bond in molecular systems.

Test data exam-vb42 is an example.

The code developed by the group of Marco Nascimento at Instituto de Química Laboratório de Química Teórica e Modelagem Molecular, Departamento de Físico-Química. Universidade Federal do Rio de Janeiro has been added to the VB2000 GAMESS version. Our thanks go to Marco for agreeing to allow the code to be widely used and to David de Sousa from Marco’s Group for assistance. David sent upgrades which were included in the code in June 2018. Specific documentation is available in the DOC folder.

Data for AIMPAC in PUNCH file for VB function.

The GAMESS directive AIMPAC=.TRUE. can now give data for the VB function. It also has an option to print the natural orbitals of the VB function.

The code changes are in parley.src. Since it checks for a VB2000 run first, it does not give AIMPAC data for the method used in GAMESS prior to VB2000, i.e. RHF, ROHF, etc. There are various variables passed to parley by COMMON statements.

The directive $PRINTNOS, along with AIMPAC=.TRUE. In the GAMESS data, also allows the printing of the natural orbitals of the VB function.

IRC or surface scan runs.

Related to the first two developments at the top of this document is the prospect of doing a high level internal reaction coordinate (IRC) run from the transition state down to the reactants or products and then calculating a VB function at each point. This is what Peter Karadakov and David Cooper have done in several papers, but not all in one run. They get a reaction path using MP2 or higher methods and then do SC runs at a set of points on that path. This is now implemented in the GAMESS/VB2000, as are the very similar surface scans.

Test data exam-vb37 calculates a VB saddle point starting from the HF saddle point.
Test data exam-vb38 does a VB calculation at each point of an IRC run.
Test data exam-vb39 does a VB calculation at each point of a scan.

Molden input

The Molden file format provides a useful interchange between different programs. We have now allowed this for VB2000 in the stand-alone version. Molden files from a variety of programs, including ORCA, CFOUR and PSI4, have been tested. The program molden2aim, designed to produce *.wfx and *.wfn files from Molden files, also cleans up Molden files and should be tried if you find a Molden file that does not work.

In the stand-alone version, VB2000 can read the geometry (the [Atoms] block in the Molden file), basis set (the [GTO] block) and Hartree-Fock orbitals (the [MO] block) from a Molden file. The use of a Molden file is obtained adding:-

```
$READMOLDEN
file_name
```

in the data. The file name should be in the same directory as the input file for the stand-alone version. This significantly reduces the work in preparing VB2000 data, allows basis sets that are not built into VB2000 provided they only use s, p and d functions, and allows that the work, used in a different system to converge to the correct function required, is not wasted. The Hartree-Fock method used in the other program can be RHF, ROHF or UHF and can be replaced by other methods such as DFT as now allowed for the GAMESS version. While the basis set is defined entirely from the Molden file, the VBOLIB is selected from the basis name in the command line. This name can be different from the basis set itself but must be a subset. Thus the basis set from the Molden file could be, for example, 6-31++G(2d2p), with 6-31G as the VBOLIB selected from the name in the command line. In choosing a VBOLIB in this way one has to remove polarisation functions entirely before removing diffuse functions, so in the case of 6-31++G(2d2p), only 6-31G can be selected for the VBOLIB as there is no VBOLIB for the 6-31++G or 6-31+G basis sets.

This was suggested by a user who had to change from GAMESS to a different program because of convergence difficulties with particular DFT methods for his molecules of interest.

Only the [Atoms], [GTO] and [MO] blocks in the Molden file are read, except for the non-standard [Title] block which consists of a single line title. If found this is output by VB2000. The Molden file has to start with the single line - [Molden Format]. While the case in these directives in defined in the Molden format, case is ignored by VB2000. Please report any Molden files that do not work with VB2000.

The ability to use a VBOLIB different from the basis set name discussed above has been introduced for normal VB2000 runs. This is done by a new directive:-

```
$NEWLIB
basis_name
```

The VBOLIB named “VBOLIBbasis_name” will then be used rather than the VBOLIB for the basis name (or “GEN”) in the command line. Note that “VBOLIBbasis_name” has to match a VBOLIB name exactly. So basis_name = “6-31GSTAR” is needed, and “6-31G*” is not allowed.

In the GAMESS version a different approach has been taken. A program called molden2gamvb is released with VB2000. It uses a Molden file as input and creates a GAMESS input file, with a very bare bones $VB2000 group. The GAMESS input can use GUESS=SKIP in the $GUESS group, and avoid calculating the HF orbitals by $SKIPGMSHF. Note that both of these must be present or neither. This input file contains the geometry, and basis set in a $DATA group, a $GUESS group with GUESS=SKIP as above, $SKIPGMSHF in the $VB2000 group, and the Hartree-Fock orbitals in a VEC group. The $CONTRL group and the $VB2000 group have to be edited. A $EIGENVALUE … $END block is created containing the HF orbital energies. If this is absent, GAMESS/VB2000 will set them to zero. It is useful to have the orbitals energies. There are other programs and scripts that create a GAMESS input file from a Molden file. One is a python script, gamess2molden.py. These do not produce the EIGENVALUE … $END block.

Test data exam-vb40 is an input file mostly generated by a molden2gamess program that includes the HF orbitals so they are skipped in GAMESS.

Ionicity or seniority numbers.

Structures can now be selected based on the equivalent ionicity (I) or seniority number (S). Ionicity is the number of pairs using the same orbital and seniority is the number of single occupied orbitals. Thus spin coupled for benzene is ionicity 0 or seniority number 6. I and S are related by S = M – S/2 or I = (M – S)/2 where M is the number of electrons. This feature allows calculations between spin coupled and CASVB.

Test data exam-vb41 tests a calculation with structures selected by ionicity or seniority numbers.

Eigenvalues of Overlap printed.

The overlap matrix S over the initial guess VB orbitals and the final VB orbitals are printed along with the eigenvalues of S, which gives a clear indication of whether near linear dependencies are present in the orbitals. The overlap matrix of the VB structures and its eigenvalues can also be printed to investigate linear dependencies in the structures.

Weights.

The Inverse Overlap Weights by Gallup and Norbeck are now reported along with three other weights in earlier versions.

Changes to centroids of charge and extents.

A matrix is now printed for distances between the centroids. Small changes have been made to make the Gaussian and GAMESS centroid code the same

Checking input.

SUBROUTINE CHKINPUT uses a list of all known directives to report directives that are not in the list and are thus spelling errors. The directives so found are reported close to the top of the VB2000 output. A GAMESS EXETYP=CHECK run now uses CHKINPUT to give more information on VB2000 directives found. VB directives can now be written using lower case letters.

Other new directives

$XYZREACTION is a variation of $XYZFILE that gives a multiple set of coordinates from a REACTION run.

$PUNCHS punches overlaps that can then be used to compared VBOS from different runs with a small program called compare.f.

$READVEC reads an initial guess from a GAMESS like $VEC file.

$VECONLY outputs a file in the GAMESS like $VEC file format of the VBOs. Note that $PLTORB in GAMESS/VB2000 does this among other things, hence the “ONLY” in the name.

The directives $VECONLY, $VECMOS, $VECLMOS and $VECINIT add their output as several $VEC .. $END blocks to a single file with extension *.vec. Thus the *.vec file is something like a VB2000 equivalent of the GAMESS *.dat file. These files, whether generated by the stand-alone, GAMESS or Gaussian versions, in the GAMESS like $VEC file format can be read by MacMolPlt. This is very useful for improving the initial guess and generally debugging the input and understanding the output from VB2000. Note that Brett Bode, the author of MacMolPlt, has added support for VB2000. The menu for orbitals that have been imported from $VEC groups files now includes “VB2000 molecular orbitals”, “VB2000 initial guess”, “VB localised molecular orbitals” and “VB2000”. VB2000 fixes the line immediately above the “ $VEC” line to be read by MacMolPlt to get VB2000 support in the menu. Many thanks to Brett for this and other support.

Stopping on Nan.

In cases of poorly defined problems or very bad initial guesses, VB2000 can give “NaN” for the energy at any macroiteration. Once this occurs it continues and can waste valuable computer time. VB2000 now stops if NaN occurs in this way. There are compile flags for both gfortran and ifort that are supposed to do this. However, I can not get them to work and they may have to be for the compiling of every module. The solution is that NaN can be defined as neither greater than or equal to zero, or less than or equal to zero. A procedure STOPNAN checks this and terminates the run.

$DENSCUBE

$DENSCUBE has been extended, so far only for the GAMESS version, to produce a CUBE file of the GAMESS HF density. The density is actually the density for the method used in GAMESS, but has really only been tested when that method is HF. In conjunction with the CUBE file for the total VB density, this can be used to generate a CUBE file for the density difference and a program, get_cube_diff.f, is provided to do just that.

Spin coupled.

While no major improvements have been made to the spin coupled code, there has been an investigation into how many electrons can be used in the spin coupled code. For a singlet state, it appears that SC(12) is the highest. SC(n,m) has more structures than SC(n), so the limit there is likely to be lower. For odd-electron systems the number of structures for doublet SC(n-1) is equal to that for singlet SC(n), so doublet SC(11) may be possible. SC(14) is not possible, but close to the cusp. If the code were made parallel and improved, and then run on a faster machine, it might run. Currently we estimate it may take about 100 days on an Intel i7 computer.

The problem is largely the very rapid increase in the number of structures, which his shown below for doublet odd and singlet even systems:-

No of electrons.	No of structures.	No of electrons.	No of structures.
5, 6	5	19, 20	16796

7, 8	14
9, 10	42
11, 12	132
13, 14	429
15, 16	1430
17, 18	4862
21, 22	58786
23, 24	208012
25, 26	742900
27, 28	2674440
29, 30	9694845

It is suggested that SC calculations be started from the orbitals from a simpler calculation. For example if the PP-GVB structure is expected to be the dominant one, one could start from a VB(2n) calculation with one structure (removing the strong orthogonality condition from the n pairs). Intermediate between this single structure and SC(2n) is to optimise the coefficients of the structures and not the orbitals. This is achieved using SC(2n) in the command line along with CIONLY. This is significantly faster than the full SC which optimises both the coefficients and the orbitals. However, the energy is higher than the full SC. SC(14) with CIONLY is possible while full SC as we explain above is not. We are grateful to David Sousa for pointing out the advantage of this use of CIONLY. Using CIONLY as an initial guess may also speed up full SC.

This should be be put into the context of other work. Using his CASVB code in MOLPRO, David Cooper (personal communication) has done SC(14) but it took a lot of effort and computer time. Jeppe Olsen has recently published a new method (J. Chem. Phys. **143**, 114102, 2015) reporting a SC(18) calculation for the chromium trimer. He reported in 2016(personal communication) that the present limit is SC(20), but hoped to get up to SC(24) in 2016 and SC(30) in 2017. We are not aware that he has achieved this. This will be an amazing achievement if he does.

<u>Different compilers.</u>

Most testing has been done with gfortran up to version 5.2. gfortran 5.3 has been tested under cygwin in Windows 7. Since February 2018, gfortran 7.2.0 was used successfully for the stand-alone and GAMESS versions. ifort has been used less frequently to version 12.1.5, but gives an error on one test run that is being investigated. The stand-alone version can be compiled with the most recent (2013) version of the G95 Fortran compiler. G95 can not compile GAMESS, because of inadequate support for READ*16, but it helped to find some minor bugs in other subroutines.

<u>Licensing issues.</u>

VB2000 is now issued under the terms of the GNU General Public License. A notice to this effect is at the top of each module. References to the former license in vb2000.src have been removed.

<u>Minor changes and bug fixes.</u>

In both vb2000.src and vb2gms.src, several routines, as I needed to study them, have been updated by adding “IMPLICIT NONE, declaring all variables and improving indentation. Unfortunately this leads to a lot of entries in the diff file that do not represent major changes. Bug fixes are detailed below.

Bug fixes:-

A problem with the dot notation for SC.
REACTOR runs in GAMESS version for UNITS=BOHR were incorrect.
Failure of centroid code for HF only runs.

Supplementary programs

Several programs including compare.f mentioned above are in the VB2000/TOOLS directory. See the comments in the programs and the README file for more information.