

# **VGW Molecular Module**

**Jason Deckman**  
Mandelshtam Group, UC Irvine

e-mail: [jason.deckman@gmail.com](mailto:jason.deckman@gmail.com)

# 1 Overview

The VGW Molecular module can be used to model any molecular system within the Variational Gaussian Wavepacket formalism, with the condition that all inter- and intra- molecular potentials are represented as a linear combination of Gaussians, of the form given by 1. The purpose of the code is to derive equilibrium structural and thermodynamic properties in systems where quantum effects are significant. The code was written in FORTRAN 90 and the principle module is found in the file “**vgwmolecular.f90**”. Additional modules were written for applications with Monte-Carlo simulations, namely for Parallel-Tempering replica exchange Monte Carlo in the NPT and NVT ensembles, with accompanying files for the administration of the program. These files, along with a brief description are:

## **vgwmolecular.f90**

Contains the code necessary for the implementation of the VGW molecular method within a parallel framework for simulation of molecular clusters and bulk materials in the NPT/NVT ensembles.

## **vgwmain.f90**

The main module for implementing and running the parallel Monte-Carlo simulation using the VGW molecular module above. Here simulation parameters are read in from the input file “**inputf.dat**” (specified later), initializations are performed and the program is run.

## **vgwmc.f90**

The module containing the code for carrying out the Monte-Carlo steps.

## **vgwpt.f90**

The code for implementing the replica exchange monte-carlo in parallel - performs the swapping between adjacent nodes.

## **vgwobs.f90**

Used for computing and outputting observables to file, along with other data.

In all files the code is sufficiently commented for explaining of all variables and subroutines.

## 2 VGW Molecular module

The principle file is the molecular module mentioned above and its utilization is discussed here in detail. For an explanation of all variables see the actual code in ‘**vgwmolecular.f90**’. Before going into detail, a conceptual discussion of the VGW molecular framework is first given.

Calculations using the VGW formalism within the Mandelshtam group have primarily involved single particles, except for a study involving water clusters (*para*-hydrogen was also investigated but was treated as a point particle). Molecules are more difficult to simulate because internal degrees of freedom are present, which contribute significantly to the physical properties, and further add to the computational burden. Additionally, the intra-molecular interactions are different in nature from the van der Waals interactions. They involve a different set of potentials and frequently more than one, e.g. water, that can be more complex.

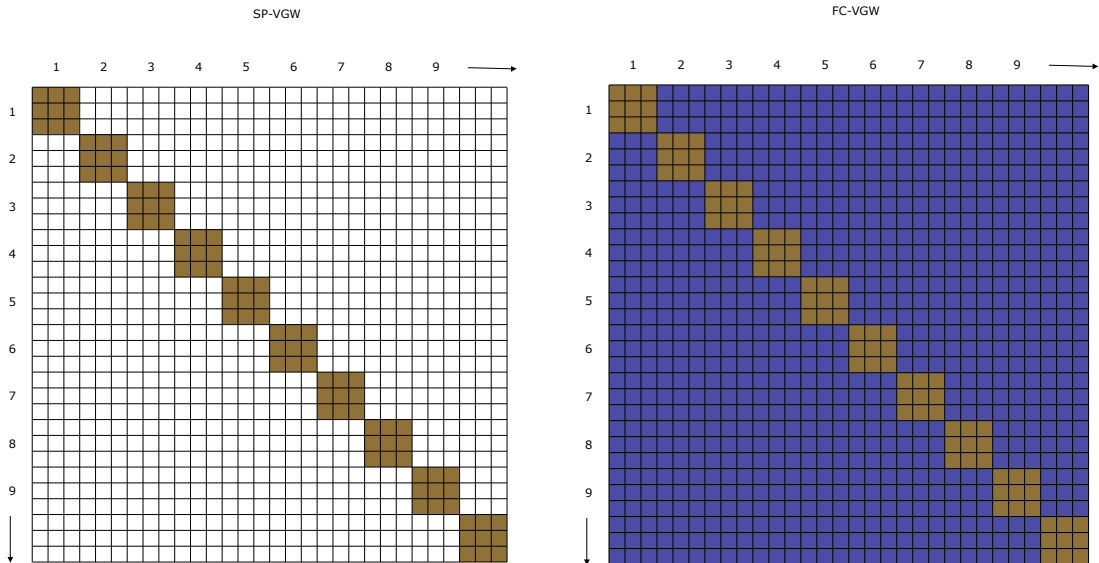


Figure 1: The Gaussian width matrix  $G$  pertaining to the SP-VGW (left) and the FC-VGW (right). In the former only the block diagonal elements are computed where in the latter all elements are calculated, where the off-diagonal components correspond to the inter-particle coupling.

As the name suggests, the VGW molecular formalism adapts the VGW method to the more complicated molecular framework. In theory, if all the molecular interactions in a system can be defined by pair-wise atomic interactions in the form of Gaussians (given by eqn. 1), then it can be modeled within the VGW formalism. To achieve this the VGW molecular is written in such a way that all intra- and inter- molecular potentials are defined in terms of atomic, pairwise interactions, and for every interaction type a set of Gaussian parameters is prescribed.

$$U(r_{ij}) \approx \sum_{k=1}^n c_k \exp(-\alpha_k r_{ij}^2) \quad (1)$$

The adaptation of the VGW method to handle molecular degrees of freedom is relatively straight forward, the only major change being in the way the  $\mathbf{G}$  matrix is constructed and the augmentation needed to account for the different pair-wise interactions. The VGW molecular format is essentially a hybrid between the single-particle VGW (SP-VGW) and the fully coupled (FC-VGW) form. In the former, the width matrix is block diagonal, consisting of  $N$   $3 \times 3$  blocks, each pertaining to a single particle (or atom) (1, left). Only the diagonal block elements of the  $\mathbf{G}$  matrix are computed and propagated, leaving the off-diagonal correlation elements zero. This results in better scaling ( $N^2$ ) but diminished

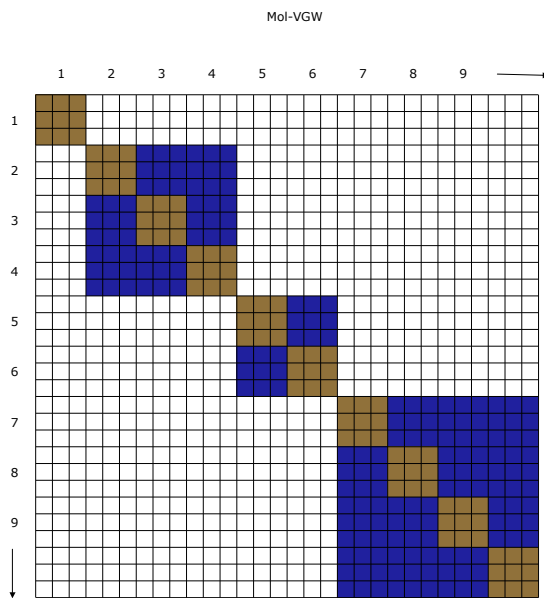


Figure 2: The Gaussian width matrix  $\mathbf{G}$  pertaining to the molecular form of the VGW, where the blocks correspond to individual molecules and all elements within are computed. These elements correspond only to intra-molecular interactions and those that are intermolecular are treated the same as those in the SP-VGW.

accuracy. In the FC-VGW, all elements are computed, which yields greater accuracy at the cost of an increased  $N^3$  scaling (1, right). In the VGW molecular formalism, the  $\mathbf{G}$  matrix consists of  $m$  molecular blocks of size  $3N \times 3N$ , where  $N$  is the number of atoms in each molecule 2. The off-diagonal elements of each molecular block are computed if the pair interaction is intra-molecular; for inter-molecular interactions, only the  $3 \times 3$  diagonal blocks pertaining to each particle are computed. Thus, the molecular blocks are considered fully coupled while the rest of the matrix is null. If every particle were to be assigned to the same molecule we would be using the FC-VGW and a full  $3N \times 3N$   $\mathbf{G}$  matrix, and the opposite being true for the SP-VGW. This allows for greater accuracy for the more

complex intra-molecular interactions while having the better scaling associated with the block diagonal form. In addition to restructuring the  $\mathbf{G}$  matrix, the molecular information

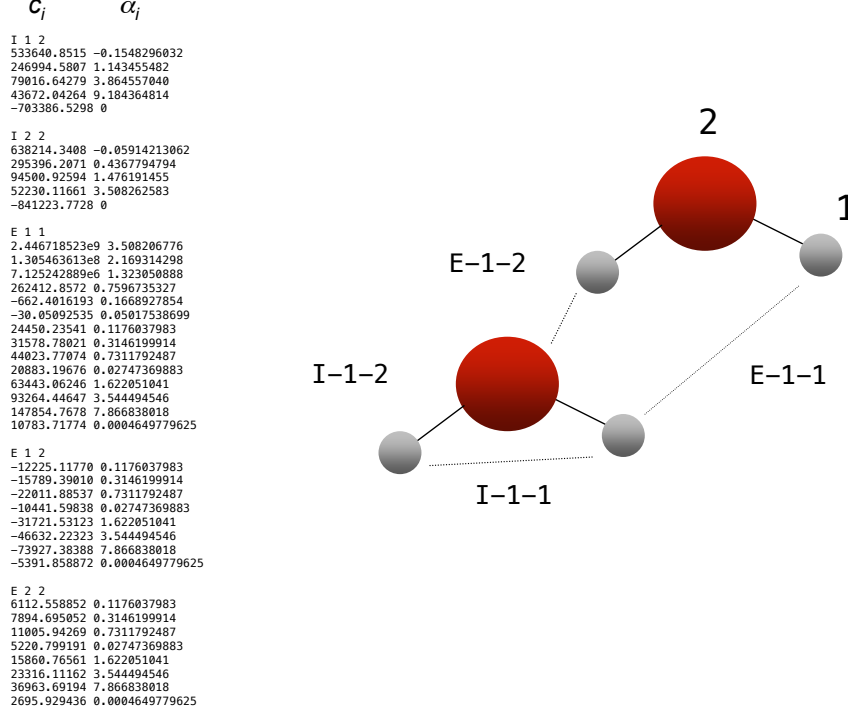


Figure 3: The labeling scheme within the molecular VGW used to identify the different type of molecular interactions and the associated set of Gaussians

and that of the nature of interaction is incorporated. This is done by assigning to each atom in the simulation a molecular label and an atom-type identifier. Then, for each pair  $ij$  in the calculation, the interaction type  $I$  (intra- or inter- molecular) was determined followed by the atomic identifiers  $A1$  and  $A2$ . The corresponding set of Gaussians pertaining to the sequence  $I-A1-A2$  was selected and used for the VGW calculations. Seen in figure Fig. 3 is the representation of this for a simulation of water. The hydrogen atoms receive a label of “1” and the oxygen a “2”. Intermolecular interactions are given the label “E” while “I” is used for the intra-molecular types. Thus, if the  $ij$ th pair in the simulations was between an H and an O on the same molecule, the interaction would be I-1-2, and the corresponding set of Gaussians on the left side of the figure would be selected. However, for convenience and simplicity, all labels are numeric, so the interaction between type 1 and 2 for an intra-molecular interaction could be 1-1-2, as will be shown ahead.

The remaining change involves the way in which the VGW density  $\rho(q_o, \beta)$  was computed. The calculation of  $\rho(q_o, \beta)$  involves taking the determinant of  $\mathbf{G}$  as shown in eqn. 2.

As the width matrix in the SP-VGW form is block diagonal, this is simply the product of  $N$   $3 \times 3$  determinants. For this new form, the latter becomes the product of the  $m$  molecular determinants each of size  $3N \times 3N$ , where each is computed first by LHA decomposition of the molecular block followed by taking the product of the diagonal elements. Then all the molecular determinants are multiplied yielding that of the entire  $\mathbf{G}$  matrix.

$$\rho(\mathbf{q}; \beta) = \langle \mathbf{q}, \beta | e^{-\beta \hat{H}} | \mathbf{q}, \beta \rangle \quad (2)$$

$$= \langle \mathbf{q}, \beta/2 | \mathbf{q}, \beta/2 \rangle \quad (3)$$

$$= 4\pi^{\frac{3N}{2}} ||G'||^{-\frac{1}{2}} \exp[2\gamma(\beta/2)]. \quad (4)$$

## 2.1 The Input Files

The VGW molecular module requires two input files that are read in upon initialization with subroutine “INITIALIZE VGWM( $\dots$ )”, which are ‘**vgwdata**’ and ‘**gauss.dat**’. The former contains the molecular information, e.g. what molecule each atom belongs to and what type of atom it is. The latter contains the sets of the Gaussian parameters used for the potentials pertaining to each possible unique pair interaction.

In ‘**vgwdata**’ Each row pertains to one atom of the molecule and corresponds to the same row and atom in the coordinate file - thus it is important to maintain the same ordering between to the files. The file contains four columns, the first contains the atomic mass, a decimal number), the second the molecule number the atom belongs to, an integer, and the third column is the atom type identifier, also and integer. The fourth column is a label used for identification purposes and isn’t used within the program - only for clarity when reading the file.

‘**vgwdata**’ layout:

```
<atomic mass >    <molecule number>    <atom type>    <label>
.
.
.
(# atoms)
```

Any two atoms with the same id number are thus considered part of the same molecule. Atoms belonging to the same molecule must be ordered consecutively row-wise - the rows

```

16.0 1 1 0
1.0 1 2 H
1.0 1 2 H
16.0 2 1 0
1.0 2 2 H
1.0 2 2 H
16.0 3 1 0
1.0 3 2 H
1.0 3 2 H
16.0 4 1 0
1.0 4 2 H
1.0 4 2 H
16.0 5 1 0
1.0 5 2 H
1.0 5 2 H
16.0 6 1 0
1.0 6 2 H
1.0 6 2 H
16.0 7 1 0
1.0 7 2 H
1.0 7 2 H
16.0 8 1 0
1.0 8 2 H
1.0 8 2 H

```

Figure 4: Sample ‘**vgwdata**’ file for 8 water molecules

must be grouped together, each having the same mol number. It follows that, that the coordinates in the coordinate file are also grouped together by molecule.

Before performing a simulation, all possible atom types must be identified and given a number accordingly, which then goes into the third column. For example a hydroxyl hydrogen on an ethanol molecule could be given the identifier ‘3’ where a hydroxyl H in benzyl alcohol could be given a ‘7’. In the fourth column (which is strictly for user identification and thrown away in the code) the labels pertaining to the latter two could be ‘H-OEth’ and ‘H-OBen’.

In ‘**gauss.dat**’ the gaussian parameters for each possible pair interaction are grouped in successive, adjacent blocks with no spacing between. The very first row though is reserved for the total number of blocks. For example, water has five total different pair interactions, so the number here would be 5, followed by each of the five blocks. The first row in each block contains four integers which are used to identify the set. The first integer is either 1 or 2, where 1 denotes an *intra*- and 2 an *inter*- molecular interaction. The next two numbers are the atom type identifiers found in column 3 of the ‘**vgwdata**’ file and the parameters to follow describe the interaction between this pair type. Using the hydroxyl labels just described, the first three values for the interaction between two hydroxyl H’s of ethanol, would be ‘2 3 3’. The last value denotes the total number of gaussians (the value  $n$  in eqn.

1 above) in the rows to follow. The remaining  $n$  rows contains two columns of data, the first being for the coefficient  $c_k$  and the second for the exponent  $a_k$ .

‘**gauss.dat**’ layout:

```
< # of sets of Gaussians (N) >
< 1 or 2 >   < atom type 1 >   < atom type 2 >   < n Gaussians >
< c1 >      < α1 >
< c2 >      < α2 >
.
.
.
< cn >      < αn >
.
.
.
(N sets of Gaussians)
```

Upon initialization the Gaussian parameters above are read in to the four dimensional array `GDATA(2*MAXGAUSS+1, MAXTYPE, MAXTYPE, 2)` and the four numbers on the line preceding the parameters are the indices used to access the coefficients and exponents in the array. The array is of fixed size and can be augmented by changing the constant parameters `MAXGAUSS` and `MAXTYPE`, where `MAXGAUSS` is the maximum number of gaussian ( $c_k$ ,  $\alpha_k$ ) pairs of parameters and `MAXTYPE` is the maximum number of atom types. Currently in the program both values are set at 20.

### 3 Additional moldules

The ‘**vgwmolecular.f90**’ contains the necessary code for quantum calculations of molecular systems within the VGW formalism and can be readily used by outside programs. As mentioned above though, several modules are also available that implement a parallel VGW-Molecular Monte-Carlo simulation in the NVT/NPT ensembles. In Fig. 6 below is a makefile that can be used to compile the entire program with the command ‘make’ on the command line. Note that the LAPACK libraries `-llapack` and `-lblas` are also required for linking.

The main module requires the input file ‘**inputf.dat**’ to initialize the program parameters. The sample code which shows the variable names into which the data is read can be found in Fig. 7. A comment on each line contains a brief description of each and a more detailed description of each follows, followed by a sample ‘**inputf.dat**’ file in Fig. 8:

**TMIN**: The lowest temperature (K) to be used for the Monte-Carlo run. This is converted to inverse temperature ( $\beta$ ) and molecular VGW is propagated to this temperature



```

5
1 1 2 5
533640.8515 -0.1548296032
246994.5807 1.143455482
79016.64279 3.864557040
43672.04264 9.184364814
-703386.5298 0
1 2 2 5
638214.3408 -0.05914213062
295396.2071 0.4367794794
94500.92594 1.476191455
52230.11661 3.508262583
-841223.7728 0
2 1 1 14
2.446718523e9 3.508206776
1.305463613e8 2.169314298
7.125242889e6 1.323050888
262412.8572 0.7596735327
-662.4016193 0.1668927854
-30.05092535 0.05017538699
24450.23541 0.1176037983
31578.78021 0.3146199914
44023.77074 0.7311792487
20883.19676 0.02747369883
63443.06246 1.622051041
93264.44647 3.544494546
147854.7678 7.866838018
10783.71774 0.0004649779625
2 1 2 8
-12225.11770 0.1176037983
-15789.39010 0.3146199914
-22011.88537 0.7311792487
-10441.59838 0.02747369883
-31721.53123 1.622051041
-46632.22323 3.544494546
-73927.38388 7.866838018
-5391.858872 0.0004649779625
2 2 2 8
6112.558852 0.1176037983
7894.695052 0.3146199914
11005.94269 0.7311792487
5220.799191 0.02747369883
15860.76561 1.622051041
23316.11162 3.544494546
36963.69194 7.866838018
2695.929436 0.0004649779625

```

Figure 5: Sample ‘**gauss.dat**’ file for a simulation of water molecules. Five sets are required, two for the intra- (O-H, H-H) and three for the inter- (H-H, O-O, H-O) molecular interactions

```

FC =mpif90
FFLAGS = -O3
#FFLAGS = -g -O0
OBJECTS = vgwmolmain.o vgwmolecular.o vgwmolmc.o vgwmolobs.o vgwmolpt.o dlsode.o
LIBS = -llapack -lblas

all: vgwmolmc.exe

vgwmolmc.exe: $(OBJECTS)
    $(FC) $(FFLAGS) -o $@ $^ $(LIBS)
    cp $@ ~/exc

vgwmolecular.o: vgwmolecular.f90
    $(FC) -c -o $@ $(FFLAGS) $^

vgwmolmc.o: vgwmolmc.f90
    $(FC) -c -o $@ $(FFLAGS) $^

vgwmolmain.o: vgwmolmain.f90
    $(FC) -c -o $@ $(FFLAGS) $^

vgwmolobs.o: vgwmolobs.f90
    $(FC) -c -o $@ $(FFLAGS) $^

vgwmolpt.o : vgwmolpt.f90
    $(FC) -c -o $@ $(FFLAGS) $^

vgwmolmc.o : vgwmolecular.o

clean:
    $(RM) $(OBJECTS) vgwmolmc.exe

.PHONY: clean

```

Figure 6: The makefile to be used for compiling the VGW molecular parallel Monte-Carlo program for simulations in the NVT/NVT ensembles

```

OPEN(UNIT=8,FILE='inputf.dat',STATUS='OLD')
  READ(8,*) TMIN           ! Minimum temperature for PTMC
  READ(8,*) TMAX           ! Maximum temperature " "
  READ(8,*) PMIN           ! Minimum Pressure
  READ(8,*) PMAX           ! Maximum Pressure
  READ(8,*) NSTEP          ! Number of MC steps
  READ(8,*) BLX            ! Cube length (for PBC)
  READ(8,*) SL             ! Step length
  READ(8,*) ATOL           ! Tolerance parameter for dlsode (1e-5 rec.)
  READ(8,*) WF             ! Reset # accepted moves and scale step length after WF steps
  READ(8,*) VGWFUF        ! VGW observables update frequency
  READ(8,*) WRF            ! VGW write freq.
  READ(8,*) RCON           ! Constraining radius
  READ(8,*) GRIDSIZE       ! Tau grid size
  READ(8,*) EDG            ! Equidistant temperature grid (non geometric)? (1=yes)
  READ(8,*) SWPINT         ! Swap interval
  READ(8,*) TAG            ! Atom name label (for VMD)
  READ(8,*) VWRF           ! VMD coords write freq
  READ(8,*) CONT           ! Continue previous MC run (1 = yes)
  READ(8,*) RCUTOFF        ! Cutoff radius
  READ(8,*) ENOT           ! E zero
  READ(8,*) VLEN           ! Volume step length
  READ(8,*) VSFREQ         ! Attempt volume change frequency (after every n MC steps)
  READ(8,*) BLMAX          ! Maximum box length
  READ(8,*) NVOUT          ! Volume output frequency
CLOSE(8)

```

Figure 7: Code for reading in data from the input file from ‘vgwmolmain.f90’

during quenching. If run in parallel this is the temperature of the head node (rank 0).

**TMAX:** The highest MC temperature (K). In a parallel run the highest ranked node is assigned and quenched to this temperature. The temperatures of the other replicas are assigned temperatures between the this and the lowest temperature, by either geometric series or using an equidistant grid.

**PMIN:** The minimum pressure for parallel MC runs at different pressures and constant temperatures. If this is non-negative, then it will be assumed the simulation is at constant T and the replicas span a range of pressures, where the head node is assigned PMIN as it was for TMIN for constant P. TMAX then becomes the value for the constant temperature.

**PMAX:** The maximum pressure. If the simulation is constant temperature ( $PMIN > 0$ ) the highest ranked node is assigned this pressure, otherwise this becomes the pressure for all replicas for constant pressure simulations. Pressure is in reduced Lennard-Jones units ( $k_B K / \text{\AA}^3$  units).

**NSTEPS:** The total number of Monte-Carlo steps. This includes all step types: translational, volume change, and PT swapping.

**BLX:** The box length (angstroms,  $\text{\AA}$ ) of the cube for periodic boundary conditions. If this is negative, the PBC will be turned off and a cluster will simulated.

**SL:** The step length for translational steps.

**ATOL:** The tolerance parameter for absolute error used by `dlode.f` for the integration of the differential equations. This is the absolute error between successive time steps. The smaller the value the higher the accuracy with the tradeoff of diminished speed and vice-versa.

**WF:** Step scaling frequency. The steps are scaled every WF MC steps according to the acceptance ratio ( $\# \text{ ACCPT}$ )/WF. After scaling, the number of accepted steps is reset and the count is begun again.

**VGWUF:** Update frequency for VGW observables, namely the partition function. Every VGWUF MC steps, the partition function is updated by adding the most recent density  $\rho(\beta)$  to  $Z$ , the partition function.

**WRF:** The frequency at which observables are written to a text file. Every WRF MC steps, the partition function and volume data is written to file.

**RCON:** The constraining radius for non PBC simulations (clusters). If MC translational moves move coordinates outside of RCON, the step is rejected.

**GRIDSIZE:** The gridsize for the temperature grid for each node. The temperature of each node spans a range of GRIDSIZE equidistant temperatures starting with the node temperature up to the higher temperature of the adjacent node.

**EDG:** If EDG is set to 1, then the replica temperatures are distributed equidistantly, otherwise they are distributed geometrically.

**SWPINT:** The swap interval for parallel tempering. Neighboring nodes exchange replicas every SWPINT MC steps.

**TAG:** Text label for writing X-Y-Z atomic coordinates to used by VMD., e.g. “Ne” for Neon, “H” for hydrogen, etc.

**VWRF:** The frequency at which the VMD coordinates are written; every VFRW MC steps.

**CONT:** Continue the previous run (1 = yes)? If set to 1, then the state files pertaining to each node are read in at runtime and therefore must be present. Additionally, the directory ‘latestcoords’ must be present with the latest coordinate files ‘lcoord.<node>’ pertaining to each node.

**RCUTOFF:** The cutoff distance for the potential between pairs of atoms. If  $r_{ij} > \text{RCUT-}$

OFF than the interaction is ignored and doesn't become a part of the energy.

**ENOT:** An arbitrary energy value for adding to the density in the form of  $\exp(-\beta \times ENOT)$  to help offset large densities that the machine may not be able to accommodate.

**VLEN:** The volume step length for volume MC steps. This is the initial value and is subsequently scaled to achieve an acceptance ratio range of 35-45%.

**VSFREQ:** The frequency at which volume steps are made; every VSFREQ MC steps a volume change is attempted (if an NPT simulation - if this value is negative or the box length is negative than an NVT simulation is assumed and no volume steps are made).

**BLMAX:** The constraining box length for an NPT simulation - any volume (box length) change MC step that exceeds this is rejected.

**NVOUT:** Volume output frequency (not currently used).

A sample input file for a simulation of bulk *para*-hydrogen:

13.5	% T min
23.5	% T max
-1	% P min
0.0072098	% P max
10000000	% # MC steps
10.00	% Cube length
0.05	% Initial step length
1e-5	% ATOL parameter for dlsode
100	% Reset frequency for acceptance ratio
1	% VGW observable update frequency
10000	% Observable write frequency, after n updates
12.5	% Constraining radius for MC run
25	% Grid size
0	% Equidistant grid (non-geometric)? (1=yes)
100	% PT swap interval
Ne	% VMD coords tag
10000	% VMD coords write frequency
0	% Continue previous run
10.0	% Cutoff radius
0.0	% ENOT
0.50	% Volume step length
5	% Volume step frequency (after n trans. steps)
30.0	% Maximum box length
-1	% Volume dump frequency

Figure 8: Sample ‘**inputf.dat**’ file for an NPT simulation of bulk parahydrogen.