



# DICE

Monte Carlo  
Molecular Simulation

## Manual

*Kaline Coutinho* (kaline@if.usp.br),

*Henrique Musseli Cezar* (henrique.musseli@gmail.com) and

*Sylvio Canuto* (canuto@if.usp.br)

Universidade de São Paulo

São Paulo, Brazil

Version 3.0 (2018)

Last actualization in 14/Ago/19

# SUMMARY

<b>1. SUMMARY OF THE PROGRAM CAPABILITIES.....</b>	<b>3</b>
<b>2. INSTALLATION AND EXECUTION .....</b>	<b>3</b>
<b>3. CITATION.....</b>	<b>4</b>
<b>4. INPUT INFORMATIONS .....</b>	<b>4</b>
4.1 INPUT FILE: <i>ST_INPUT</i> .....	4
4.2 EXAMPLES OF STANDARD INPUT FILE ( <i>ST_INPUT</i> ).....	8
4.2.1 <i>Thermalization</i> .....	8
4.2.2 <i>Simulation</i> .....	8
4.2.3 <i>Continuation</i> .....	8
4.3 INPUT FILE: <i>LJNAME</i> .....	9
4.4 EXAMPLES OF <i>LJNAME</i> INPUT FILE .....	10
4.4.1 <i>Water</i> .....	10
4.4.2 <i>Benzene</i> .....	11
4.4.3 <i>Benzene in water</i> .....	11
4.4.4 <i>Furfural in water (Thermodynamic Perturbation Theory)</i> .....	12
4.5 INPUT FILE: <i>OUTNAME.CM</i> .....	13
4.6 EXAMPLE OF <i>OUTNAME.CM</i> INPUT FILE .....	13
4.7 INPUT FILE: <i>OUTNAME.XY</i> .....	13
4.8 EXAMPLE OF <i>OUTNAME.XY</i> INPUT FILE .....	14
4.9 OUTPUT INFORMATIONS .....	15
4.9.1 <i>Output file: st_output</i> .....	15
4.9.2 <i>Output file: outname.avr</i> .....	15
4.9.3 <i>Output file: outname.dat and outname.res</i> .....	16
4.9.4 <i>Output file: outname.gr</i> .....	16
4.9.5 <i>Output file: outname.xyz</i> .....	16
<b>5. PROGRAM STRUCTURE .....</b>	<b>17</b>
<b>6. ERROR MESSAGES AND WHAT TO DO.....</b>	<b>18</b>
<b>7. THEORY.....</b>	<b>20</b>
7.1 SITE-SITE INTERACTION POTENTIAL ENERGY .....	20
7.2 LONG RANGE CORRECTION OF THE POTENTIAL ENERGY .....	21
<b>8. CONVERTING UNITS.....</b>	<b>22</b>

Observation: NOT IMPLEMENTED OPTIONS ARE WRITTEN IN RED COLOR.

## 1. Summary of the Program Capabilities

This code performs simulations of pure liquids and liquids composed by different types of molecules, using the Metropolis Monte Carlo method (MC). All molecules have rigid geometry. The total number of molecules  $N$  and the temperature  $T$  are fixed, but either the volume  $V$  or the pressure  $P$  can fluctuate during the simulation. One of the  $NVT$  or  $NPT$  ensembles can be chosen. The intermolecular atomic interaction is described by the sum of the Lennard-Jones and the Coulomb potentials. The border effects are avoided through the use of the image method with periodic boundary conditions. This method allows each molecule to interact with an infinite number of molecule images. However, considering the computational limits and the inclusion of periodicity in the system, each molecule is restricted to interact up to a cutoff radius. In the way that, each molecule sees either the molecule or the respective image, not simultaneously both. In the energy calculation, the error caused by the cutoff radius is reduced with the calculation of the long range correction (LRC) of the potential. The LRC of the Lennard-Jones potential is evaluated using the pair radial distribution function and the LRC of the Coulomb potential is evaluated using the reaction field method in dipole interactions. One MC step is defined after attempt to move  $N$  molecules randomly selected. During the simulation, basically four quantities are evaluated and accumulated: the radial distribution function (RDF), the potential energy,  $U$ , the first and second virial function,  $W$  and  $\Theta$ , respectively. The atomic coordinates are also storage in a file with XYZ format. After the simulation, the thermodynamic properties are calculated using the fluctuation theory and many informations of the system are printed in the output files, including configurations of the system that can be easily used for further studies of solvent effects by quantum mechanical calculations.

## 2. Installation and Execution

The DICE program is now available for download in the address [http://www ???](http://www.???).

It can be compiled using any UNIX or WINDOWS FORTRAN/77 compiler.

To runs it, use the following command line in the UNIX or DOS windows:

```
dice < st_input > st_output
```

Finding any difficulty in the DICE installation, contact us. We appreciate all the help we can get to make sure that the program runs on different platforms and to find bugs. We require that all work published which is based on use of the DICE program cite it by the reference bellow.

The authors of the program accept no responsibility for the performance of the code, nor the correctness of the results obtained.

### 3. Citation

H. M. Cezar, S. Canuto K. Coutinho, DICE version3.0: A Monte Carlo program for molecular liquid simulation, Universidade de São Paulo, Brazil (2018).

H. M. Cezar, S. Canuto, K. Coutinho, “Solvent Effect on the Syn/Anti Conformational Stability: A Comparison Between Conformational Bias Monte Carlo and Molecular Dynamics Methods”, *Int. J. Quantum Chem.* **119**, 25688 (2019). DOI: 10.1002/qua.25688

### 4. Input Informations

DICE needs two input files: *st\_input* and *ljname* and use a third *outname.cm* or *outname.xy* optionally.

#### 4.1 Input file: *st\_input*

This is read in the logical unit 5 (*st\_input* = standard input). In this file, the lines, that start with #, are understood as comments and the information are associated with keywords.

The general keywords are:

- **acceptance =** values between 0.0 and 1.0 that control the acceptance ratio of new configurations.

**(Default: acceptance = 0.5)**

- **accum =** **yes** - all the properties accumulated in a previous simulation (saved in *outname.res* file) are added in the present simulation. In fact, the present simulation is a continuation of the previous and the following message is printed in the *st\_output* file: '\*\*\* THIS RUN IS A CONTINUATION \*\*\*'  
**no** - the present simulation neglects all the properties accumulated in a previous simulation.

**(Default: accum = no)**

- **box =**  $L_x$   $L_y$   $L_z$   $L_{zeffective}$  This option is used only if **geom = 1**.  $L_{zeffective}$  should be  $\leq L_z$  and is used only in the initial configuration to distribute the solvent molecules in a small z axis (used simulations with surfaces).
- **coolstep =** number of steps where the Metropolis sampling technique is not used, only configurations that lower the energy are accepted.

**(Default: coolstep = 0 if init = no or coolstep = 150 otherwise.)**

- cutoff =    yes           - the present simulation neglects the interaction energy between molecules that are separated by distances larger than the cut-off-radius ( $r_c = L_{\min}/2$ ).
- no           - disable the cut-off-radius.

**(Default: cutoff = on)**

- dens = density in  $\text{g/cm}^3$ .
- dielectric = dielectric continuum constant. If this keyword is not declared, then the long range correction of the Coulomb potential will not be evaluated (NOT WORKING !!).
- geom =    0           - a cubic box is used.
- 1           - a parallelogramic box is used.
- 2           - a rhombic dodecahedron box is used (NOT WORKING !!).

**(Default: geom = 0)**

- init =    yes           - a random configuration is generated.
- yesread    - a configuration is generated with the center-of-mass positions and orientations read in the *outname.cm* file. If the file has less positions and orientations than the number of total molecules then the missed informations are generated randomly.
- yesreadxyz - a configuration is generated with the atomic coordinates read in the *outname.xy* file. If the file does not have the coordinates of all molecules then the missed informations are generated randomly.
- yesord     - a ordered configuration is generated. Each direction will have the cubic root of the total number  $N$  of molecules equally separated. (This option only works if  $N$  has an exact cubic root.)
- no         - a configuration is read in the *outname.dat* file.

**(Default: init = no if the outname.dat file exists or init = yes otherwise.)**

- iprint = interval of MC steps that informations, like configuration energy and acceptance ratio, are printed in the *st\_output* file.

**(Default: iprint = 1)**

- irdf = interval of MC steps between configurations used in the calculation of the RDF.

**(Default: irdf = 5 if init = no or irdf = 0 otherwise)**

- isave = interval of MC steps that all data are printed in the *outname.dat*, *outname.res* and *outname.xyz* files.

**(Default: isave = 1000)**

- `ljname` = name of the input file with molecule informations, like geometry and potential (see format next).
- `mstop` =
 

0	- normal simulation.
1	- the first molecule does not move during the simulation.
-1	- the first molecule does not move during the simulation and it will have an atomic periodic boundary condition. It should be used for very large solute or surfaces.
A	- any other number will impose that the first A molecules do not move during the simulation.

**(Default: `mstop` = 0)**

- `nmol` = number of molecules. If more than one type of molecules is used in the simulation, i.e.  $m$  types of molecules are used, then `nmol` is read in the format ( `nmol(i)`,  $i=1, m$ ).
- `ncores` = number of cores to run in parallel.
- `nratio` = interval of MC steps that is used to adjust the acceptance ratio of molecular movements.
- `nstep` = number of MC steps in the *NVT* ensemble. In the *NPT* ensemble, `nstep` means the number of molecular moves between the volume changes.

**(Default: In the *NVT* ensemble, `nstep` = 50000 if `init` = no or `nstep` = 3000 otherwise. In the *NPT* ensemble `nstep` = 5)**

- `outname` = name of the output files. Four output files are generated with the same name and different terminations: `outname.dat`, `outname.gr`, `outname.res` and `outname.xyz`. Their contents are described in the output section.
- `perturbation` =
 

yes	- the Thermodynamic Perturbation Theory is used. With this option the <code>ljname</code> file such be modified.
no	- normal simulation.

**(Default: `perturbation` = no)**

- `press` = pressure in atm. It is used only in the *NPT* ensemble simulation.

**(Default: `press` = 1)**

- `rbuf` =
- `sampling` = the value of the cavity radius to perform the preferential sampling. Molecules in side the cavity will be attempted to move two times more often than the molecules out side the cavity (technical details:  $N_{in}/N_{eff} \approx 0.5 \Rightarrow p = 0.38$ ).
- `seed` = seed of the random number generator.

- temp = temperature in Kelvin.  
(Default: temp = 298.15)
- title = title
- vratio = interval of MC steps that is used to adjust the acceptance ratio of volume changes.  
(Default: In the *NVT* ensemble, vratio = 0. In the *NPT* ensemble vratio = 50)
- vstep = number of volume changes. If vstep = 0, the *NVT* ensemble is selected, otherwise the *NPT* ensemble is selected.  
(Default: vstep = 0)  
(Default: nratio = 50 if init = no or nratio = 10 otherwise.)
- upbuf = interval of MC steps that the neighbor list is actualized (Suggestion: upbuf = 360).
- biasphi =
- equiphi =
- flex =
- fudgeij =
- fudgeicb =
- lambdac =
- pcbmc =
- ntrialphi =
- nsf =
- savefraglib =
- sfint =

Summary of keywords:

Keywords that MUST be declared ALWAYS:

```
ljname =
outname =
seed =
upbuf =
ncores =
$end
```

Keywords that MUST be declared IF the initial configuration does not exist (*outname.dat*) or init = yes or yesread:

```
nmol =
dens =
```

Default values:

```
geom = 0
temp = 298.15
press = 1
accum = no
vstep = 0
iprint = 1
isave = 1000
```

If the initial configuration ( <i>outname.dat</i> file)	
exists	does not exist
init = no	init = yes
coolstep = 0	coolstep = 150
nstep = 50000	nstep = 3000
irdf = 5	irdf = 0
iratio = 50	iratio = 10

```
vratio = 0
ncores
```

## 4.2 Examples of standard input file (*st\_input*)

### 4.2.1 Thermalization

Input to generate the initial configuration (randomly) of 343 water molecules and to perform a NVT simulation (thermalization). The informations about the water molecule are read in the file `h2o` and the output is written in the files: `h2o343.dat`, `h2o343.res` and `h2o343.xyz`. By default: i) the system is cooled in 150 steps, ii) 3000 MC steps are performed and iii) no RDF is calculated.

```
#
title = NVT-simulation of 343 water molecules (thermalization)
ljname = h2o
outname = h2o343
seed = 1091
ncores = 2
upbuf = 4
#
init = yes
nmol = 343
dens = 1.00
#
$end
```

### 4.2.2 Simulation

Input to read the initial configuration in the file `h2o343.dat` from a previous simulation and to perform a NVT simulation with 70000 x 343MC steps. The informations about the water molecule are read in the file `h2o` and the output is written in the files: `h2o343.dat`, `h2o343.res`, `h2o343.gr` and `h2o343.xyz.1` (if the previous `h2o343.xyz` was not deleted). By default no property is accumulated from the previous simulation.

```
#
title = NVT-simulation of 343 water molecules (equilibrium)
ljname = h2o
outname = h2o343
ncores = 2
upbuf = 4
#
nstep = 70000
$end
```

### 4.2.3 Continuation

Input to continue the previous simulation with more 15000 x 343MC steps.

```
title = NVT-simulation of water (equilibrium - continuation)
ljname = h2o
outname = h2o343
ncores = 2
upbuf = 4
```



```
#
nstep = 15000
accum = yes
$end
```

### 4.3 Input file: **ljname**

The second input file is declared in the keyword **ljname**. This file has information about the molecules and each line should have a specific information, as described below:

- line 1: combination rule of the Lennard-Jones  $\sigma$  parameter,  
 ( rule = \*  $\Rightarrow \sigma_{ij} = \sqrt{\sigma_i \sigma_j}$  or +  $\Rightarrow \sigma_{ij} = (\sigma_i + \sigma_j) / 2$  ).
- line 2: number of different types of molecule (**m**).
- line 3: number of atoms of the type 1 of molecules, (**na(1)**).
- following **na(1)** lines: the RDF atomic label (**label**), the atomic number (**an**), the Cartesian coordinates (**x**, **y**, **z**) in Å, the atomic charge (**q**) in unit of *e* and the Lennard-Jones potential parameters  $\epsilon$  in kcal/mol and  $\sigma$  in Å.
- following lines: repeat the informations to the molecular type 2, 3 and so on, if they exist.

Format of the **ljname** file:

```
rule
m
na(1)
label1      an1      x1      y1      z1      q1      ε1      σ1
label2      an2      x2      y2      z2      q2      ε2      σ2
  ⋮          ⋮          ⋮          ⋮          ⋮          ⋮          ⋮          ⋮
labelna(1)  anna(1)  xna(1)  yna(1)  zna(1)  qna(1)  εna(1)  σna(1)
na(2)
label1      an1      x1      y1      z1      q1      ε1      σ1
label2      an2      x2      y2      z2      q2      ε2      σ2
  ⋮          ⋮          ⋮          ⋮          ⋮          ⋮          ⋮          ⋮
labelna(2)  anna(2)  xna(2)  yna(2)  zna(2)  qna(2)  εna(2)  σna(2)
  ⋮
$end
```

**Important:** The atomic number 104 corresponds to a dummy atom, i.e., an atom without mass.

If in the simulation, there is more than one type of molecule, then the type 1 is treated as the solute. It makes some differences: (i) related to the ordering of the configurations saved in the outname.xyz file, the center-of-mass of the solute molecule (or molecules) is placed in the origin of the box and the molecular coordinates are printed distance-wise from the origin. (ii) related to the radial distribution file, the two functions  $G_{\text{CM-CM}}(r)$  and  $G_{\text{Rmin-Rmin}}$  are evaluated between solute-solvent molecules.

If the Thermodynamic Perturbation Theory is used, this file changes a little. A new column is added after the Lennard-Jones parameter. This column is a number that multiplies the atomic mass, allowing an atom to disappear or to transform itself in other. Additionally, after the solvent information, an comment line and informations of the 2 perturbed molecules should be included. See the file format below:

rule								
m								
na(1)		(solute informations, example: dihedral angle $\phi$ )						
label <sub>1</sub>	an <sub>1</sub>	x <sub>1</sub>	y <sub>1</sub>	z <sub>1</sub>	q <sub>1</sub>	$\epsilon_1$	$\sigma_1$	mass scale <sub>1</sub>
label <sub>2</sub>	an <sub>2</sub>	x <sub>2</sub>	y <sub>2</sub>	z <sub>2</sub>	q <sub>2</sub>	$\epsilon_2$	$\sigma_2$	mass scale <sub>2</sub>
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
label <sub>na(1)</sub>	an <sub>na(1)</sub>	x <sub>na(1)</sub>	y <sub>na(1)</sub>	z <sub>na(1)</sub>	q <sub>na(1)</sub>	$\epsilon_{na(1)}$	$\sigma_{na(1)}$	mass scale <sub>na(1)</sub>
na(2)		(solvent informations)						
label <sub>1</sub>	an <sub>1</sub>	x <sub>1</sub>	y <sub>1</sub>	z <sub>1</sub>	q <sub>1</sub>	$\epsilon_1$	$\sigma_1$	
label <sub>2</sub>	an <sub>2</sub>	x <sub>2</sub>	y <sub>2</sub>	z <sub>2</sub>	q <sub>2</sub>	$\epsilon_2$	$\sigma_2$	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
label <sub>na(2)</sub>	an <sub>na(2)</sub>	x <sub>na(2)</sub>	y <sub>na(2)</sub>	z <sub>na(2)</sub>	q <sub>na(2)</sub>	$\epsilon_{na(2)}$	$\sigma_{na(2)}$	
⋮								
		Double-wide perturbation (comment line)						
na(m+1)		(first perturbed solute informations, dihedral angle $\phi+\delta$ )						
label <sub>1</sub>	an <sub>1</sub>	x <sub>1</sub>	y <sub>1</sub>	z <sub>1</sub>	q <sub>1</sub>	$\epsilon_1$	$\sigma_1$	mass scale <sub>1</sub>
label <sub>2</sub>	an <sub>2</sub>	x <sub>2</sub>	y <sub>2</sub>	z <sub>2</sub>	q <sub>2</sub>	$\epsilon_2$	$\sigma_2$	mass scale <sub>2</sub>
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
label <sub>na(1)</sub>	an <sub>na(1)</sub>	x <sub>na(1)</sub>	y <sub>na(1)</sub>	z <sub>na(1)</sub>	q <sub>na(1)</sub>	$\epsilon_{na(1)}$	$\sigma_{na(1)}$	mass scale <sub>na(1)</sub>
na(m+2)		(second perturbed solute informations, dihedral angle $\phi-\delta$ )						
label <sub>1</sub>	an <sub>1</sub>	x <sub>1</sub>	y <sub>1</sub>	z <sub>1</sub>	q <sub>1</sub>	$\epsilon_1$	$\sigma_1$	mass scale <sub>1</sub>
label <sub>2</sub>	an <sub>2</sub>	x <sub>2</sub>	y <sub>2</sub>	z <sub>2</sub>	q <sub>2</sub>	$\epsilon_2$	$\sigma_2$	mass scale <sub>2</sub>
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
label <sub>na(2)</sub>	an <sub>na(2)</sub>	x <sub>na(2)</sub>	y <sub>na(2)</sub>	z <sub>na(2)</sub>	q <sub>na(2)</sub>	$\epsilon_{na(2)}$	$\sigma_{na(2)}$	mass scale <sub>na(1)</sub>
\$end								

#### 4.4 Examples of ljname input file

##### 4.4.1 Water

Input to simulate pure water with SPC potential. To calculate the RDF, the atomic labels (in first column) are used. In this example, there are two atomic labels: label 1 = oxygen (O) and label 2 = hydrogen (H). Therefore, three pairwise RDF are calculated:  $G_{O-O}(r)$ ,  $G_{O-H}(r)$  and  $G_{H-H}(r)$  (in this case, remember that  $G_{O-H}(r)$  and  $G_{H-O}(r)$  are accumulated together). In addition more two  $G(r)$  are always calculated: between center of masses  $G_{CM-CM}(r)$  and between closest atoms of different molecules  $G_{Rmin-Rmin}(r)$ .

*								
1								
3								
1	8	0.0000	0.0000	0.0000	-0.82	0.155	3.165	
2	1	0.5774	0.8165	0.0000	0.41	0.000	0.000	

```

2      1  0.5774  -0.8165  0.0000  0.41  0.000  0.000
$end

```

In the second column the atomic number are shown:  $na\ 1 = 8$  for oxygen and  $na\ 2$  and  $na\ 3 = 1$  for hydrogen. In the columns 3, 4 and 5 the Cartesian coordinates are shown in Å. In columns 6, 7 and 8 the potential parameters are shown:  $q$  in unit of elementary  $e$ ,  $\epsilon$  in kcal/mol and  $\sigma$  in Å.

#### 4.4.2 Benzene

Input to simulate pure benzene with the 6-site potential. In this example, there are two different atomic labels: label 1 = carbon (C) and label 2 = hydrogen (H) and label 3 = dummy atoms at  $2\text{Å}$  up and down the ring (XX). These dummy atoms do not interfere in the simulation. It is used only in the RDF calculation. With these labels, the pairwise RDF calculated are:  $G_{C-C}(r)$ ,  $G_{C-H}(r)$ ,  $G_{C-XX}(r)$ ,  $G_{H-H}(r)$ ,  $G_{H-XX}(r)$  and  $G_{XX-XX}(r)$ , additionally to  $G_{CM-CM}(r)$  and  $G_{Rmin-Rmin}(r)$  (see example 4.4.1).

```

*
1
13
1      6  0.000000  1.4000  0.000  0.000  0.110  3.750
1      6  1.212436  0.7000  0.000  0.000  0.110  3.750
1      6  1.212436 -0.7000  0.000  0.000  0.110  3.750
1      6  0.000000 -1.4000  0.000  0.000  0.110  3.750
1      6 -1.212436 -0.7000  0.000  0.000  0.110  3.750
1      6 -1.212436  0.7000  0.000  0.000  0.110  3.750
2      1  0.000000  2.4881  0.000  0.000  0.000  0.000
2      1  2.154671  1.2440  0.000  0.000  0.000  0.000
2      1  2.154671 -1.2440  0.000  0.000  0.000  0.000
2      1  0.000000 -2.4881  0.000  0.000  0.000  0.000
2      1 -2.154671 -1.2440  0.000  0.000  0.000  0.000
2      1 -2.154671  1.2440  0.000  0.000  0.000  0.000
3      104 0.000000  0.0000  2.000  0.000  0.000  0.000
3      104 0.000000  0.0000 -2.000  0.000  0.000  0.000
$end

```

#### 4.4.3 Benzene in water

Input to simulate benzene (type 1 = solute) in water (type 2 = solvent). In this example, there are three types of RDF: between solute-solute molecules (if there is more than one solute molecule), between solute-solvent molecules and between solvent-solvent molecules. Also in this example, there is two atomic labels for the solute: label 1 = carbon (C) and label 2 = hydrogen (Hb); and two atomic labels for the solvent: label 1 = oxygen (O) and label 2 = hydrogen (Hw). Therefore, the pairwise RDF between solute-solute molecules are:  $G_{C-C}(r)$ ,  $G_{C-Hb}(r)$  and  $G_{Hb-Hb}(r)$ ; between solute-solvent molecules are:  $G_{C-O}(r)$ ,  $G_{C-Hw}(r)$ ,  $G_{Hb-O}(r)$ ,  $G_{Hb-Hw}(r)$ ; and between solvent-solvent molecules are:  $G_{O-O}(r)$ ,  $G_{O-Hw}(r)$  and  $G_{Hw-Hw}(r)$ .

```

*
2
12

```

1	6	0.000000	1.4000	0.000	0.000	0.110	3.750
1	6	1.212436	0.7000	0.000	0.000	0.110	3.750
1	6	1.212436	-0.7000	0.000	0.000	0.110	3.750
1	6	0.000000	-1.4000	0.000	0.000	0.110	3.750
1	6	-1.212436	-0.7000	0.000	0.000	0.110	3.750
1	6	-1.212436	0.7000	0.000	0.000	0.110	3.750
2	1	0.000000	2.4881	0.000	0.000	0.000	0.000
2	1	2.154671	1.2440	0.000	0.000	0.000	0.000
2	1	2.154671	-1.2440	0.000	0.000	0.000	0.000
2	1	0.000000	-2.4881	0.000	0.000	0.000	0.000
2	1	-2.154671	-1.2440	0.000	0.000	0.000	0.000
2	1	-2.154671	1.2440	0.000	0.000	0.000	0.000
3							
1	8	0.000000	0.0000	0.0000	-0.820	0.155	3.165
2	1	0.577400	0.8165	0.0000	0.410	0.000	0.000
2	1	0.577400	-0.8165	0.0000	0.410	0.000	0.000
\$end							

#### 4.4.4 Furfural in water (Thermodynamic Perturbation Theory)

Input to simulate furfural (type 1 = solute) in water (type 2 = solvent) using the Thermodynamic Perturbation Theory with changes in the  $\varphi(\text{O-C-C=O})$  dihedral angle. In this example, the furfural is simulated with  $\varphi = 9^\circ$  and perturbed to  $\varphi = 0^\circ$  (type 3 = solute) and  $\varphi = 18^\circ$  (type 4 = solute).

*							
2							
11	(Furfural with angle= 9)						
1 8	-0.66506	-1.19317	0.00000	-0.121417	0.140	2.90	1.00
2 6	0.00000	0.00000	0.00000	0.088547	0.070	3.55	1.00
2 6	-0.90239	1.04672	0.00000	-0.225334	0.070	3.55	1.00
2 6	-2.19359	0.45113	0.00000	-0.113554	0.070	3.55	1.00
2 6	-1.98881	-0.90853	0.00000	-0.040323	0.070	3.55	1.00
3 1	-0.65371	2.10013	0.00000	0.154926	0.030	2.42	1.00
3 1	-3.15148	0.95246	0.00000	0.128679	0.030	2.42	1.00
3 1	-2.65367	-1.76025	0.00000	0.152592	0.030	2.42	1.00
2 6	1.46100	0.00000	0.00000	0.393730	0.105	3.75	1.00
4 8	2.16113	-0.98681	-0.15630	-0.406947	0.210	2.96	1.00
5 1	1.90540	1.00140	0.15861	-0.010900	0.030	2.42	1.00
3 0							
1 8	0.000000	0.0000	0.0000	-0.820	0.155	3.165	
2 1	0.577400	0.8165	0.0000	0.410	0.000	0.000	
2 1	0.577400	-0.8165	0.0000	0.410	0.000	0.000	
#Perturbed geometries							
11	(Furfural with angle= 0)						
1 8	-0.66506	-1.19317	0.00000	-0.113742	0.140	2.90	1.00
2 6	-0.00000	0.00000	0.00000	0.093111	0.070	3.55	1.00
2 6	-0.90239	1.04672	0.00000	-0.229589	0.070	3.55	1.00
2 6	-2.19359	0.45113	0.00000	-0.092507	0.070	3.55	1.00
2 6	-1.98881	-0.90853	0.00000	-0.066166	0.070	3.55	1.00
3 1	-0.65371	2.10013	0.00000	0.154373	0.030	2.42	1.00
3 1	-3.15148	0.95246	0.00000	0.123161	0.030	2.42	1.00
3 1	-2.65367	-1.76025	0.00000	0.160697	0.030	2.42	1.00
2 6	1.46100	0.00000	0.00000	0.378572	0.105	3.75	1.00
4 8	2.16113	-0.99911	0.00000	-0.403259	0.210	2.96	1.00
5 1	1.90540	1.01388	0.00000	-0.004650	0.030	2.42	1.00
11	(Furfural with angle= 18)						
1 8	-0.66506	-1.19317	0.00000	-0.126960	0.140	2.90	1.00

2	6	0.00000	0.00000	0.00000	0.097045	0.070	3.55	1.00
2	6	-0.90239	1.04672	0.00000	-0.218960	0.070	3.55	1.00
2	6	-2.19359	0.45113	0.00000	-0.131439	0.070	3.55	1.00
2	6	-1.98881	-0.90853	0.00000	-0.031840	0.070	3.55	1.00
3	1	-0.65371	2.10013	0.00000	0.153364	0.030	2.42	1.00
3	1	-3.15148	0.95246	0.00000	0.134313	0.030	2.42	1.00
3	1	-2.65367	-1.76025	0.00000	0.150622	0.030	2.42	1.00
2	6	1.46100	0.00000	0.00000	0.385450	0.105	3.75	1.00
4	8	2.16113	-0.95021	-0.30874	-0.401587	0.210	2.96	1.00
5	1	1.90540	0.96426	0.31331	-0.010007	0.030	2.42	1.00
\$end								

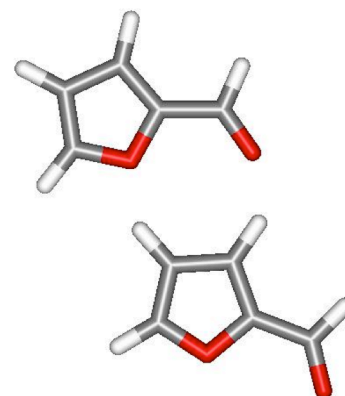
#### 4.5 Input File: *outname.cm*

The third input file is optional and is used as an alternative to the random procedure for generation the initial configuration. In this file, the initial position and orientation of  $n$  molecules are informed, where  $n$  is chosen by the user. The position is defined as the Cartesian coordinates (x, y, z) of the center-of-mass with origin in the center of the box used in the simulation. The orientation is defined with a selection of an axis (1= x, 2= y and 3= z) and an angle in degree. The format of this file is:

X <sub>1CM</sub>	Y <sub>1CM</sub>	Z <sub>1CM</sub>	Axis_number	Angle <sub>1</sub>
X <sub>2CM</sub>	Y <sub>2CM</sub>	Z <sub>2CM</sub>	Axis_number	Angle <sub>2</sub>
X <sub>3CM</sub>	Y <sub>3CM</sub>	Z <sub>3CM</sub>	Axis_number	Angle <sub>3</sub>
⋮	⋮	⋮	⋮	⋮
X <sub>nCM</sub>	Y <sub>nCM</sub>	Z <sub>nCM</sub>	Axis_number	Angle <sub>n</sub>
\$end				

#### 4.6 Example of *outname.cm* input file

Input to simulate 2 furfural molecules (type 1 = solute) in water (type 2 = solvent). In this example, only the initial positions and orientations of the 2 furfural molecules are informed (shown on the right-hand side). Therefore, the water molecules will be placed randomly in the box.



0.00	0.00	0.00	0	0.00
2.00	5.00	0.00	3	-25.00
\$end				

#### 4.7 Input File: *outname.xy*

The fourth input file is optional also and is used as an alternative to the random procedure for generation the initial configuration. In this file, the initial atomic coordinates (x, y, z) of  $n$  molecules are informed, where  $n$  is chosen by the user. The format of this file is:

```

x1      y1      z1
x2      y2      z2
x3      y3      z3
⋮        ⋮        ⋮
xm      ym      zm
$end

```

```

0.31798   -3.07530   -0.08460
1.36866   -2.20253   -0.10188
0.90499   -0.90071   -0.08762
-0.51387  -0.99021   -0.06001
-0.81359  -2.33215   -0.05938
1.51700   -0.00802   -0.09617
-1.22610  -0.17700   -0.04277
-1.74086  -2.88655   -0.04328
2.73091   -2.72975   -0.13080
3.02317   -3.91418   -0.14121
3.51115   -1.94456   -0.14309
-1.45102   1.43489   -0.06193
-0.78596   2.62806   -0.06193
-1.68835   3.67478   -0.06193
-2.97955   3.07919   -0.06193
-2.77477   1.71953   -0.06193
-1.43967   4.72819   -0.06193
-3.93744   3.58052   -0.06193
-3.43963   0.86781   -0.06193
0.67504    2.62806   -0.06193
1.37517    1.62895   -0.06193
1.11944    3.64194   -0.06193
$end

```

#### 4.8 Example of *outname.xy* input file

This is the same example shown previously, input to simulate 2 furfural molecules (type 1 = solute) in water (type 2 = solvent). In this example, only the atomic coordinates of the 2 furfural molecules are informed. Therefore, the water molecules will be placed randomly in the box. Note that there are 22 lines in this file (11 for

each furfural) and the atomic identification follows the sequence defined in the file *lname* (O, C, C, C, C, H, H, H, C, O, H) shown below. The standard file is also shown. It generates the initial configuration by reading the atomic coordinates of the 2 furfural molecules in the file *frf2w.xy* and randomly placing the 600 water molecules. After 150 steps of cooling, 5000x5x600 MC steps are performed in the NPT ensemble, with the 2 furfural molecules in stopped positions. In this example, the water molecules are thermalized around the furfural dimer. After the thermalization, the keyword *mstop* can be deleted of the input file, then all the 602 molecules will be free to translate and rotate.

```

*
2
11 (furfural with CHELPG charges)
1 8 -0.66506 -1.19317 0.00000 -0.113742 0.140 2.90 1.0
2 6 -0.00000 0.00000 0.00000 0.093111 0.070 3.55 1.0
2 6 -0.90239 1.04672 0.00000 -0.229589 0.070 3.55 1.0
2 6 -2.19359 0.45113 0.00000 -0.092507 0.070 3.55 1.0
2 6 -1.98881 -0.90853 0.00000 -0.066166 0.070 3.55 1.0
3 1 -0.65371 2.10013 0.00000 0.154373 0.030 2.42 1.0
3 1 -3.15148 0.95246 0.00000 0.123161 0.030 2.42 1.0
3 1 -2.65367 -1.76025 0.00000 0.160697 0.030 2.42 1.0
2 6 1.46100 0.00000 0.00000 0.378572 0.070 3.55 1.0
4 8 2.16113 -0.99911 0.00000 -0.403259 0.210 2.96 1.0
5 1 1.90540 1.01388 0.00000 -0.004650 0.030 2.42 1.0
3 (water SPC)
1 8 0.000000 0.0000 0.0000 -0.820 0.155 3.165
2 1 0.577400 0.8165 0.0000 0.410 0.000 0.000
2 1 0.577400 -0.8165 0.0000 0.410 0.000 0.000
$end

```

```

title = 2 furf + 600 H2O NPT (therm.)
lname = frf2w.txt
outname = frf2w
init = yesreadxyz
coolstep = 150
nmol = 2 600
temp = 298.15
press = 1.00
dens = 1.00
accum = no
vstep = 5000
nstep = 5
iprint = 1
isave = 1000
irdf = 0
iratio = 10
seed = 879
mstop = 2
$end

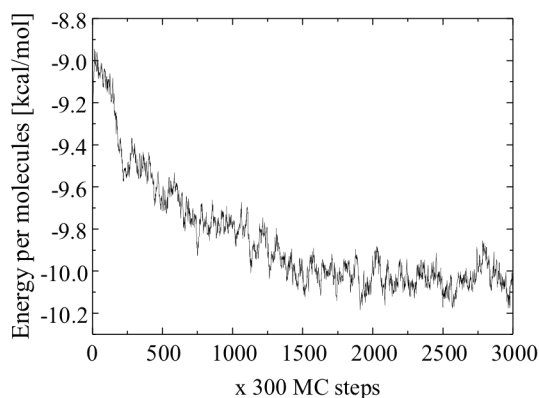
```

## 4.9 Output Informations

DICE generates five output files: *st\_output*, *outname.dat*, *outname.gr*, *outname.res*, *outname.avr* and *outname.xyz*.

### 4.9.1 Output file: *st\_output*

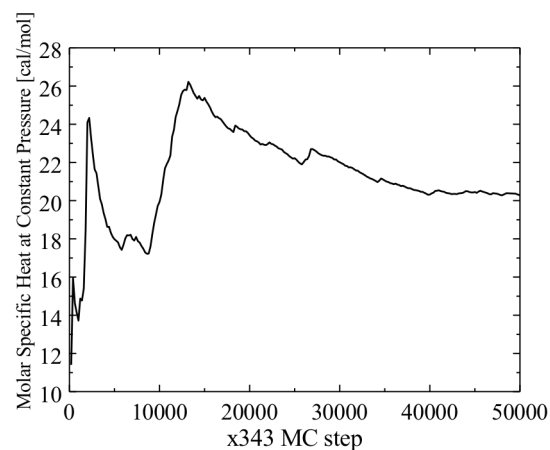
The *st\_output* file contains all informations and conditions where the simulation is performed. During the simulation, each MC step generates a configuration and instantaneous informations about this configuration are printed. These informations are: the number of the configuration, or MC step (NMOVE), the acceptance ratio (RATIO), the potential energy per molecule (U/N), the conformational entalpy per molecule (Hc/N), the potential energy of molecule 1 with the others (Us), the potential energy of the first perturbed molecule with the others (U1), the potential energy of the second perturbed molecule with the others (U2), the first (W), second virial (Theta), etc. In the end of the simulation, the averaged properties are printed. They are: the internal energy per molecule ( $\langle E/N \rangle$ ), the internal energy of molecule 1 with the others ( $\langle E_s \rangle$ ), the pressure ( $\langle P \rangle$ ), the thermal pressure coefficient (gv), the thermal expansion coefficient (ap), the isothermal compressibility (bt), the molar specific heat at constant volume (cv) and the molar specific heat at constant pressure (cp). After the simulation, the graphic of U/N versus NMOVE can check if the equilibrium has been reached. In the right-hand side, this graphic is shown as a illustration of result obtained in a simulation of 300 water molecules in the thermalization stage.



**Important:** The thermodynamic properties have their values correctly evaluated only in the equilibrium stage. If the system is in the thermalization stage these values have no physical meaning.

### 4.9.2 Output file: *outname.avr*

This file is used to store the intermediate average properties as a function of the MC step. This file can be used to analyze the convergence of the averaged properties showed in end of the *st\_output*. In the right-hand side a convergence of



the molar specific heat at constant pressure,  $c_p(\text{cal/mol})$  is shown as illustration of result obtained in a simulation of one furfural and 342 water molecules in the NPT ensemble.

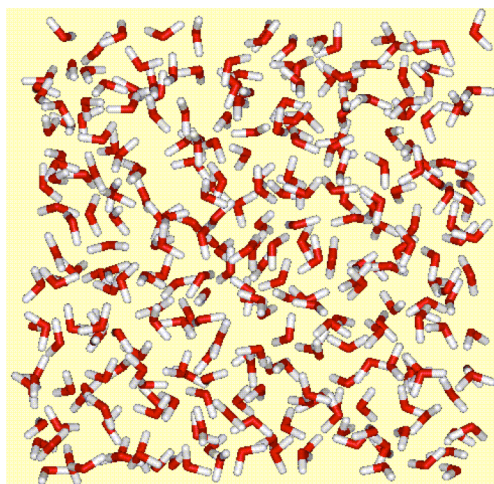
#### 4.9.3 Output file: *outname.dat* and *outname.res*

These files are used internally by the DICE to restart or continue the simulation.

#### 4.9.4 Output file: *outname.gr*

The *outname.gr* file contains the calculated RDFs and the their integration (that give the coordination numbers) as a function of the distance  $G_{i-j}(r)$  and  $N(r)$ , respectively. In this file, the first column is  $r$  the second is the  $G_{i-j}(r)$  and the third is  $N(r)$ . The RDFs are printed sequentially with respect to the atomic label (in *ljname* file). In the case of example 4.4.1, the *outname.gr* file has first

$G_{1-1} = G_{O-O}(r)$ , then  $G_{1-2} = G_{O-H}(r)$  and then  $G_{2-2} = G_{H-H}(r)$ . In the case of example 4.4.3, the solute-solute RDFs are given at first ( $G_{C-C}(r)$ ,  $G_{C-Hb}(r)$  and  $G_{Hb-Hb}(r)$ ), then the solute-solvent RDFs ( $G_{C-O}(r)$ ,  $G_{C-Hw}(r)$ ,  $G_{Hb-O}(r)$ ,  $G_{Hb-Hw}(r)$ ) and at last the solvent-solvent RDFs ( $G_{O-O}(r)$ ,  $G_{O-Hw}(r)$  and  $G_{Hw-Hw}(r)$ ).



#### 4.9.5 Output file: *outname.xyz*

The *outname.xyz* file contains ordered configurations printed in the standard XYZ format. For a liquid composed of only one type of molecule, the first (printed) molecule has its center-of-mass in the origin of the box and the following molecules are the first nearest neighbor, the second and so on. The order of the neighbors is defined by the center-of-mass-center-of-mass distance. If the liquid is composed by several types of molecules, but the type 1 (solute) has only one molecule, then the same ordering procedure is used and, in this case, the first printed molecule is the solute. However, in the case that type 1 has more than one molecule, i.e., there are several solute molecules, then the center-of-mass of the solute molecules is placed in the origin of the box and the molecules coordinates are printed distance-wise from the origin. In the figure, above one of the printed configuration in the *outname.xyz* file is illustrated with the freeware visualization program WebLab ViewerLite (<http://www.msi.com/>).



## 5. Program Structure

DICE is written in FORTRAN/77 and is composed of two files: `dice.for` and `dice.prm`. The `dice.for` file contains the main code and the subroutines. The `dice.prm` file contains the parameters that dimension the most important vectors and matrices in the program, the commons and the declarations of their variables. The parameters are:

- `mnt` = maximum number of different type of molecules.
- `mnm` = maximum number of solvent molecules.
- `mnms` = maximum number of solute molecules.
- `mna` = maximum number of atoms in each solvent molecule.
- `mnas` = maximum number of atoms in each solute molecule.
- `mbin` = maximum number of r bins in the RDF.
- `nin` = maximum number of different pairwise RDF.
- `u??` = i/o unit number.

If the values of `mnt`, `mnm`, `mnms`, `mna`, `mnas`, `mbin` and `nin` parameters are exceeded, the execution stops with an error message like "Increase the parameter `mnt`". In this case, the program should be recompiled after increasing the parameter in the file `dice.prm`.

The subroutines play specific roles in the simulation. They are:

1. Subroutine `distribution(grfil, delr, ibl, consti, iprop)`  
Normalizes and writes out the RDF functions.
2. Subroutine `energy(rxinew, ryinew, rzinew, i, rcutsq, rminsq, v12, v6, vc, overlap, vdip, dielec)`  
Calculates the pair potential energy of molecule *i* with the other molecules separated by distances smaller than the  $r_c$ .
3. Subroutine `initial(boxl, rmin, dummy)`  
Creates an initial configuration randomly.
4. Subroutine `initial_cm(cmfil, boxl, rmin, dummy)`  
Creates an initial configuration using center-of-mass position and orientation of molecules, specified in the file `outname.cm`.
5. Subroutine `initial_ord(boxl, rmin)`  
Creates an initial configuration ordered using original molecular orientation read in the `lname` file. In this case the number of molecules should be an exact cubic root (ex.: 125, 343, etc.)
6. Subroutine `input(title, mofile, namefil, inic, rest, utmp)`  
Read the input keywords related to the labels.

7. Subroutine `input_sys(utmp, tstep, igeom, temp, press, dens, seed, dielec)`  
Read the input keywords related to the initial configuration.
8. Subroutine `input_run(utmp, vstep, nstep, iprint, imean, isave, iratio, vratio)`  
Read the input keywords related to the MC simulation.
9. Subroutine `move(rxi, ryi, rzi, i, rxj, ryj, rzj, drmax, dgamax, dummy)`  
Moves the molecule *i* to a new position and orientation.
10. Subroutine `order(orm, ory, orz, list)`  
Orders a configuration.
11. Subroutine `properties(delr, ratl, li, l6i, l12i)`  
Accumulates the RDF.
12. Real function `ranf(dummy)`  
Returns a uniform random number between zero and one.
13. Subroutine `readdt(dtfil, temp, press, dens, dlmax, drmax, ibl, dielec)`  
Reads several variables and an initial configuration (normalized units) in the file `outname.dat`.
14. Subroutine `sumup(rcutsq, rminsq, overlap, v12, v6, vc, vdip, dip)`  
Calculates the potential energy for a configuration.
15. Subroutine `writcf(boxl, sni)`  
Writes a configuration in the file `outname.xyz`.
16. Subroutine `writcf_ord(boxl, sni)`  
Writes an ordered configuration in the file `outname.xyz`.
17. Subroutine `writdt(dtfil, temp, press, dens, dlmax, drmax, ibl, dielec)`  
Writes several variables and a configuration (normalized units) in the file `outname.dat`.

## 6. Error messages and what to do

- 'Error: Delete the RDF files'  
Delete the files `outname.gr.?`, where  $? = 1, 2$ , until 9.
- 'Error: Delete the configurations files'  
Delete the files `outname.xyz.?`, where  $? = 1, 2$ , until 9.
- 'Error: Need definition of sigma rule in the "ljname"'  
Declare the signal `*` or `+` in the first character of the first line of the file `ljname`.
- 'Error: Need seed to the random generator'  
Declare the keyword `seed` in the file `st_input`.

- 'Error: Need density value'  
Declare the keyword `dens` in the file `st_input`, when an initial configuration is generated.
- 'Error: Increase the parameter "mnt"'  
Edit the file `dice.prm` to increase the value of the parameter `mnt` and recompile the program.
- 'Error: Increase the parameter "mna"'  
Edit the file `dice.prm` to increase the value of the parameter `mna` and recompile the program.
- 'Error: Increase the parameter "mnas"'  
Edit the file `dice.prm` to increase the value of the parameter `mnas` and recompile the program.
- 'Error: Increase the parameter "mnm"'  
Edit the file `dice.prm` to increase the value of the parameter `mnm` and recompile the program.
- 'Error: Increase the parameter "mnms"'  
Edit the file `dice.prm` to increase the value of the parameter `mnms` and recompile the program.
- 'Error: Increase the parameter "nin"'  
Edit the file `dice.prm` to increase the value of the parameter `nin` and recompile the program.
- 'Error: Increase the parameter "mbin"'  
Edit the file `dice.prm` to increase the value of the parameter `mbin` and recompile the program.
- 'Error: The ordered initial configuration can be generated only to homogeneous liquid'  
Using the DICE program, it is not possible to generate an ordered initial configuration to liquid composed with different types of molecules.
- 'Error: Density too large'  
This error appears only when an ordered initial configuration is generated and the distance between the molecule layers is lower than the overlap distance defined in the program ( $r_{min} = 0.8 \text{ \AA}$ ).
- 'Error: "nmol" should be 64, 125, 216, 343, ...'  
This error appears only when an ordered initial configuration is generated and the number of molecules set to the simulation (`nmol`) is not an exact cubic root.

- 'Error in the outname.cm configuration. Overlap of atoms: k and l of molecules i and j'  
Change the center-of-mass positions in the file *outname.cm* or the orientation of the molecules in the *ljname* file.
- 'Error: The angle is mis-defined in outname.cm file (0 < angle < 360) '  
Change the angle in the file *outname.cm*.
- 'Error: The rotational axis is mis-defined in outname.cm file (1=X, 2=Y and 3=Z)'  
Change the axis number in the file *outname.cm*.
- 'Error in initial configuration '  
Report this error to the authors.
- 'Error: Overlap in initial configuration '  
Report this error to the authors.
- 'Error: Problem with the random generator '  
Report this error to the authors.

## 7. Theory

### 7.1 Site-site Interaction Potential Energy

DICE code uses the Lennard-Jones plus Coulomb potentials to describe the site-site interaction ( $U(r_{ij})$ ):

$$U(r_{ij}) = 4\epsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \left( \frac{q_i q_j}{r_{ij}} \right), \quad (7.1)$$

where  $r_{ij}$  is the distance between site  $i$  and site  $j$  of different molecules,  $\epsilon_{ij}$  and  $\sigma_{ij}$  are combinations of the Lennard-Jones parameters of sites  $i$  and  $j$ ,  $q_i = q_i^* (e/4\pi\epsilon_0)$  is the charge of the site  $i$  and  $(e/4\pi\epsilon_0) = 18.21(\text{\AA} \text{ kcal/mol})^{1/2}$ . The combination of the  $\epsilon$  parameter is  $\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j}$  and the  $\sigma$  parameter is  $\sigma_{ij} = \sqrt{\sigma_i \sigma_j}$  or  $\sigma_{ij} = (\sigma_i + \sigma_j)/2$  depending on the rule (\* or +) declared in the first line of the file *ljname*.

The interaction potential energy between two molecules ( $a$  and  $b$ ) is the sum of  $U(r_{ij})$  over all defined sites in molecules  $a$  (sites  $i$ ) and all defines sites in molecules  $b$  (sites  $j$ ), i.e.  $U_{ab} = \sum_{i,j} U(r_{ij})$ . The total potential energy is the sum of  $U_{ab}$  over all pairs of molecules neglecting all terms that  $r_{ab} > r_c$ , where  $r_{ab}$  is the center-of-mass distance between molecules  $a$  and  $b$  and  $r_c$  is a cut-off radius defined in the DICE as the half size of the length of the cube or the cubic root of

the system volume ( $r_c = L/2$ , where  $L = \sqrt[3]{V}$ ). Thus, the total potential energy of one configuration of the simulated system is

$$U = \sum_{a,b}^{\substack{r_{ab} \leq r_c \\ i,j}} U(r_{ij}) = \sum_{a,b}^{\substack{r_{ab} \leq r_c \\ i,j}} 4\epsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \left( \frac{q_i q_j}{r_{ij}} \right). \quad (7.2)$$

For simplification, the equation 7.2 is rewritten with three different terms due to the site-site distance power

$$U = U_{12} - U_6 + U_1 \quad (7.3)$$

where  $U_{12} = \sum_{a,b} \sum_{i,j} 4 \epsilon_{ij} (\sigma_{ij} / r_{ij})^{12}$ ,  $U_6 = \sum_{a,b} \sum_{i,j} 4 \epsilon_{ij} (\sigma_{ij} / r_{ij})^6$  and  $U_1 = \sum_{a,b} \sum_{i,j} (q_i q_j / r_{ij})$ .

Using this definition of the potential energy function, other two quantities are calculated: the first virial function,  $W$ ,

$$W = -\frac{1}{3} \left[ r \left( \frac{dU}{dr} \right) \right] = 4U_{12} - 2U_6 + \frac{1}{3}U_1 \quad (7.4)$$

and the second virial function  $\Theta$ ,

$$\Theta = \frac{1}{9} \left[ r \left( \frac{d}{dr} r \left( \frac{dU}{dr} \right) \right) \right] = 16U_{12} - 4U_6 + \frac{1}{9}U_1. \quad (7.5)$$

## 7.2 Long Range Correction of the Potential Energy

The long range corrections (LRC) of the potential energy of evaluated in two separated terms: the site-site Lennard-Jones potentials and the dipole-dipole interaction due to the Coulombic potential. The first term is obtained by the integration of the radial distribution function as the following equation:

$$U_{ij}^{\text{LRC}} = \frac{1}{2} \frac{N_i N_j}{V} \int_{r_c}^{\infty} U_{ij}(r) G_{ij}(r) 4\pi r^2 dr. \quad (7.6)$$

and  $G_{ij}(r) \approx 1$  to  $r > r_c$ . In DICE code, for distance greater than  $r_c$  the liquid will be composed by an uniform distribution of

.

.

.

UNDER CONSTRUCTION !!!

.

## 8. Converting Units

$$N_A = 6.0221367 \times 10^{23} / \text{mol}$$

$$1 \text{ cal} = 4.184 \text{ J}$$

$$1 \text{ J} = 1.439 \times 10^{20} \text{ kcal/mol}$$

$$k_B = 1.985 \times 10^{-3} \text{ kcal/mol K}$$