USER MANUAL OF THE MCMGS CODE

Last update of this document: 5 June 2025

INDEX

Author of this manual and of the mcmgs code:

Iván Cabria

Associate Professor (Profesor Titular)

Department of Theoretical Physics, Atomic Physics and Optics

University of Valladolid

ivan.cabria@uva.es

1. Introduction

The name of the code is ***mcmgs*** (Monte Carlo Method for Gas Storage). The source code is in ***Fortran***. It has been compiled and ran only under ***Linux***. As of 5 June 2025, it has been compiled, tested, and ran on seven Linux clusters or hosts.

The code mcmgs makes ***Monte Carlo simulations of a gas inside a solid porous material.*** It uses the Monte Carlo-Metropolis algorithm. A Monte Carlo simulation consists on a set of successive iterations. On each iteration the Monte Carlo-Metropolis algorithm is applied.

The code can make simulations of the NVT and ***μVT*** thermodynamical ensembles. N, V, T and μ are the number of molecules, the volume, the temperature, and the chemical potential, respectively, of the simulated system. The NVT ensemble consists of a set of molecules of a gas, inside the porous material, with constant values of N, V and T. The ***μVT*** ensemble consists of a set of molecules of a gas, inside the porous material, with ***constant values of μ, V and T.***

The NVT ensemble is also called the Canonical ensemble. The μVT ensemble is also called the Grand Canonical ensemble.

The Monte Carlo simulations of the μVT ensemble are usually called Grand Canonical Monte Carlo, GCMC, simulations.

The code can simulate the following gases: H2, CH4, N2 and CO2. The code can simulate solid porous materials made of H, O, C, Cu and Zn.

The code calculates the gravimetric and volumetric storage capacities of the solid porous material. The code calculates the storage capacities of the gas involved in the simulation. For instance, ***if the gas is hydrogen*** and the solid material is MOF-5, then the code calculates the ***hydrogen gravimetric and volumetric storage capacities*** of MOF-5.

The ***input file*** has the extension ***xyz***. There are several output files. The most important ***output file*** has the extension ***out***. In the section about the output files, we will explain the content of the .out file and the names and contents of the other output files.

2. Input

2.1 Structure of the input file

The input file has the extension xyz and has two parts or blocks.

The ***first part*** consists on the number of ***particles***, a title and the chemical symbol, coordinates and state of the particles that form the system that we want to simulate.

The order of the lines of the first part is important. The first line must be the number of particles, nparticles. The second line must be the title. The next nparticles lines must be the chemical symbol, cartesian coordinates and state of the particles.

The particles could be ***atoms or molecules***. The particles could be ***part of the solid porous material or part of the gas.***

The particles are inside a ***cell or simulation cell.***

The particles that are part of the solid porous materials will not be moved during the simulation. Their status is, therefore, fixed (f).

The particles that are part of the gas will be moved during the simulation. Their status, is, therefore, not fixed (n).

It is not mandatory to write first the lines about the particles of the solid material and then, the lines about the gas, although it is recommended. The code read mixed lines and then it sorts out the first block of the input: First, the particles of the solid material, then the particles of the gas.

All the lines of the first part or block must be in the input file.

The ***second part*** consists on a set of lines, each line containing a ***keyword and its value***. A keyword could have an ***alphanumeric*** value or a ***numeric value***. For instance:

ensemble muvt

In this instance, ensemble is the keyword, and muvt is the alphanumeric value of the keyword. Another instance:

distance 0.2

The keyword is distance, and the value is 0.2 (angstroms). This keyword has a numeric value.

The second part must be after the first part. There is not an order of the keywords inside the second part. The code ***reads the keywords in any order***. This is very convenient for the user.

It is ***not mandatory*** to write every possible keyword line in the input file. ***Some keywords have a default value.*** If one of those keywords is not written in the input file, then the code assigns to that keyword ***a default value***. This is also very convenient for the user.

It is ***mandatory*** to write the keyword line of some keywords.

The values of some keywords ***are used if another keyword has a specific value***. For instance, if the keyword geometry has the value slitpore in the input file, then the value of the keyword tuberadius will not be used and hence, the value of this keyword in the input file has not effect. If the value is tube, then the value of tuberadius will be used.

The names of the keywords were chosen to be ***self-explanatory***: The name of the keyword should represent the meaning of the keyword. For instance, distance is the keyword of the maximum distance that a molecule can be moved randomly in one iteration; maxniterations is the maximum number of iterations of the simulation.

If the value of a keyword ***is not valid***, then the code prints a message in the output file and stops. For instance, if the value of the keyword Temperature is -20, then the code prints a message and stops. The ***code checks the values of all the keywords before starting the simulation***, but there is no 100 % guarantee that the code will find every inconsistent or not valid value of the keywords.

In the second section we will see an example of input file and in the third section we will explain the meaning of the lines of the input example, line by line.

2.2 Example of input file

This is an input example of a graphene slit pore made of two graphene layers separated 12 angstroms. It is the file example.xyz. The H2 molecules are between the two layers. We have removed some lines and replaced them by …..., to avoid using too much space in this manual.

```
 96
two graphene layers separated 12 angstroms and 32 H2 molecules
C    0.000000000000000   0.000000000000000   0.000000000000000 f
C    0.000000000000000   1.420000000000000   0.000000000000000 f
…….
C    0.000000000000000   0.000000000000000   12.000000000000000 f
C    0.000000000000000   1.420000000000000   12.000000000000000 f
…….
H2   0.000000000000000   0.000000000000000   3.000000000000000 n
H2   0.000000000000000   1.420000000000000   9.000000000000000 n
……..
H2   8.608292513617320   6.390000000000000   3.000000000000000 n
H2   8.608292513617320   7.810000000000000   9.000000000000000 n
ensemble muvt
eos SRK
maxniterations 10000
nprint 500
seedoption -1
nmoleculesmovedperiteration 1
Pressure 50
Temperature 77
distance 0.2
typeofpotential1 lj
boundaryconditions periodic
percentagetrialsmovemolecules 20.0
nitersbetweenvaluesforaverages 1000
nequilibrationiterations 5000
geometry slitpore
tuberadius 4.0 0.0 0.0
innerradius 20.0
```

sphereradius 4.0 0.0 0.0

printoption all

symbolgas1 H2

typeofenergycalculation relative

cutoffradius 20.0

quantumcorrection no

nwalls 2

symbolwall C C

xwall 0.0 0.0

ywall 0.0 0.0

zwall 0.0 12.0

ncells 1 1 0

cellparameters 9.8380486 8.52 12.0

cellangles 90.0 90.0 90.0

combiningrule Good-Hope-Berthelot

typeofpotentialsg1 none

nadditionalsheets 0

symbolgas2 H2

typeofpotential2 lj

typeofpotentialsg2 none

potentialenergystep 0.1

ndepletiontemperatures 2

depletion1 0.5 300.15 0.000100 0.100000 0.5 77.00 0.000100 0.100000

In this example we can see the two parts or blocks of an input file. The first 98 lines are the first part. The remaining lines are the keyword lines and they are the second part.

2.3 Explanation of the input file, line by line

2.3.1 First part of the input example

The first line is the number of particles of the system, nparticles. In our example, 96. Valid values > 0. If nparticles is <=0, then the code prints a message and stops.

The second line is the title of the simulation. Here, any text is valid

The lines 3 to 98 are the chemical symbols, ***coordinates in angstroms*** and states of each of the 96 particles that form the system. Each line has the following structure:

Chemical symbol x coordinate y coordinate z coordinate state

For instance, the line:

C 0.000000000000000 1.420000000000000 0.000000000000000 f

means that this is a carbon atom, located at x=0, y=1.42 angstroms and z=0.0 angstroms, and it is a fixed particle (f). This carbon atom will not be moved during the simulation.

The line:

H2 8.608292513617320 7.810000000000000 9.000000000000000 n

means that this is a hydrogen molecule, located at x=8.608 angstroms, y=7.81 angstroms and z= 9 angstroms, and it is a non-fixed particle (n). Hence, this particle will be moved during the simulation.

2.3.2 Second part of the input example

In this section we present and explain, line by line, the keywords of the input file, their meaning and their valid values.

If a value of a keyword is ***not valid*** or is ***inconsistent*** with the values of another keywords, then ***the code prints a message and stops the simulation.***

ensemble muvt The type of thermodynamical ensemble. In this case, muvt, the Grand Canonical ensemble. Valid values: nvt and muvt. Default value: muvt.

eos SRK The Equation Of State (EOS) of the gas. Valid values: Shaw, SRK and Mills-Younglove. Shaw is valid only for H2 molecules. Default value: Shaw.

maxniterations 10000 The maximum number of iterations of the simulation. Valid values >= 0. Default value: 0.

nprint 500 The code will print some results every nprint iterations. Valid values >= 0. Default value: 0.

nprint 0 means that the code will not print results during the simulation, only at the end.

seedoption -1 The seed of the random numbers. It is an integer numeric value. Default value: 0. Valid values and their meaning:

< 0	No seed is used
0	Use a seed generated using random data retrieved from the operating system
1	Use /dev/urandom to generate the seed
2	Use the PID to generate the seed
3	Use the current time-of-day in milliseconds to generate the seed

nmoleculesmovedperiteration 1 Number of molecules that could be moved on a movement trial. Valid values: 0 < nmoleculesmovedperiteration < nmolecules. Default value: 1.

Pressure 50 Pressure of the system in MPa units. Valid values >= 0.1. **No default value.**

Temperature 77 Temperature of the system in K units. Valid values > 0.0. **No default value.**

distance 0.2 Maximum distance that a particle can be moved randomly in angstroms. Valid values > 0.0. Default value: 0.2. Recommended values: 0.1-1.0.

typeofpotential1 lj Type of interaction potential of the first type of gas or gas 1. Default value: lj. Valid values and their meaning:

lj	Lennard-Jones interaction potential
Lennard-Jones	Same meaning as lj
analytic	An interaction potential of the type Sum Ci zi, different from the LJ potential

constant A constant interaction potential. The value of the constant potential energy is zero.

boundaryconditions periodic Type of boundary conditions of the simulation. Valid values: free and periodic. Free means that the system has not boundary conditions, i.e., is not periodic. Default value: periodic.

percentagetrialsmovemolecules 20.0 On each iteration of a GCMC simulation there is a trial: movement, addition or deletion trial. This keyword is the probability that the trial will be a movement trial. Valid values: 0.0-100.0. Default value: 20.0. Recommended value: 20.0.

nitersbetweenvaluesforaverages 1000 The average of any physical magnitude is calculated using the value of that magnitude in the iterations nequilibrationiterations, nequilibrationiterations+nitersbetweenvaluesforaverages, nequilibrationiterations+2 nitersbetweenvaluesforaverages, etc. In this example, it will be used to calculate the gravimetric capacity and any other magnitude in the iterations 30000, 31000, 32000, etc. Valid values > 0. Default value: 100.

nequilibrationiterations 5000 After nequilibrationiterations the system is considered to be at the equilibrium or that it has converged. Valid values > 0. There is no default value.

geometry slitpore Type of geometry of the system. Default value: slitpore. Valid values and their meaning:

slitpore A 2-dimensional system. Usually consists of two planar sheets, separated a distance w. It could be periodic or non-periodic. If the system is periodic, then the cell is repeated in the xy plane.

crystal A 3-dimensional system. If the system is periodic, then the cell is repeated in the xyz space. It could be periodic or non-periodic. If the system is periodic, then the cell can be any type of cell: cubic, orthorhombic, monoclinic, triclinic, etc.

tube A 1-dimensional system. The main axis is along the z axis. It could be periodic or non-periodic. If the system is periodic, then the cell is repeated along the z axis.

sphere A non-periodic 3-dimensional system with spherical symmetry. It is not a periodic system.

torus A non-periodic 3-dimensional system with toroidal symmetry. It is not a periodic system.

cone A non-periodic 3-dimensional system with conical symmetry. It is not a periodic system.

If boundaryconditions is free, then the system is non-periodic. If boundaryconditions are periodic, then the system is periodic.

Tuberadius 4.0 0.0 0.0 Radii of the nanotubes or external radii of the tori in angstroms. Valid values, if geometry is tube > 0.0; if not, any value is valid. Default value: 0.0. The first value is the radius of the widest tube or torus. The second value is the radius of the intermediate tube or radius and the third value is the radius of the narrowest tube or radius.

innerradius 20.0 Internal radius of the torusene in angstroms. Valid values if geometry is torus > 0.0; if not, any value is valid. Default value: 0.0.

sphereradius 4.0 0.0 0.0 Radii of the fullerenes in angstroms. Valid values if geometry is sphere > 0.0; if not, any value is valid. Default value: 1.0. The first value is the radius of the widest fullerene. The second value is the radius of the intermediate fullerene and the third value is the radius of the narrowest fullerene.

printoption all The code prints also results in another files. This keyword indicates to the code which files will be printed. In this example, the value of the keyword is all. Default value: nothing. Valid values of the keyword printoption and which files are printed:

nothing or no No files are printed

intermediate intermediategeometries.xyz is printed

last lastgeometry.xyz is printed

lowest lowestgeometry.xyz is printed

lastandintermediate lastgeometry.xyz and intermediategeometries.xyz

lowestandintermediate lowestenergygeometry.xyz and intermediategeometries.xyz

lastandlowest lastgeometry.xyz and lowestenergygeometry.xyz

potentialenergy potentialenergy.dat is printed

potentialenergyandlast potentialenergy.dat and lastgeometry.xyz

potentialenergyandlowest potentialenergy.dat, lowestenergygeometry.xyz

all All the possible files are printed

The meaning and content of the files is explained in the Output section.

symbolgas1 H2 Chemical symbol of first type of gas. If there is only one type of gas, then this is its symbol. In this example, the gas is hydrogen. Valid values: H2, CH4, N2 and CO2. **No default value.**

typeofenergycalculation relative Type of calculation of the energy of the system. Valid values: Absolute and relative. Default value: relative.

In an absolute calculation all the interactions are calculated and summed up in the total energy. In a relative calculation, the total energy of the last iteration is used and only the interactions of the moved, added or removed particle are considered. Relative calculations are much faster than absolute calculations. In the iteration 0, the calculation is, obviously, an absolute calculation.

cutoffradius 20.0 The cutoff radius of the interaction potential in angstroms. If the distance between two particles (fixed or not fixed) is more than the cutoff radius, then the interaction potential is considered to be zero. Valid values > 0.0. Default value: 150.

quantumcorrection no The Feynman-Hibbs quantum corrections of the interaction potential. Valid values: no, yes and Feynman-Hibbs. The values yes and

Feynman-Hibbs have the same meaning: The Feynman-Hibbs correction is applied. Default value: no.

nwalls 2 Number of walls of the solid material. This is for slitpore geometries. Valid values > 0. Default value: 1.

symbolwall C C Chemical symbol of the atoms that form each wall. **No default value.**

xwall 0.0 0.0 x coordinate in angstrom of each wall. **No default value.**

ywall 0.0 0.0 y coordinate in angstrom of each wall. **No default value.**

zwall 0.0 12.0 z coordinate in angstrom of each wall. **No default value.**

Note: In future versions, nwalls, symbolwall, xwall, ywall and zwall will not be present in the input file. The code will analyze the coordinates and will determine the number of walls and their symbols and geometric locations.

ncells 1 1 0 The cell that contains the system will be replicated in the x, y and z directions. ncells nx ny nz means that the unit cell will be replicated nx times along x>0 and also nx times along the x<0 part. The same applies for the y and z directions. The total number of cells will be (2nx +1)(2ny+1)(2nz+1): The real cell plus (2nx +1) (2ny+1)(2nz+1) – 1 replicated or image cells. This keyword is used if the boundary conditions are periodic. Valid values of nx, ny and nz >= 0. Default values: 0 0 0.

The values of nx, ny, nz should be consistent with the geometry. The following values and geometry pairs are consistent:

slitpore	ncells nx ny 0	nx, ny > 0
crystal	ncells nx ny nz	nx, ny, nz > 0
tube	ncells 0 0 nz	nz > 0

If geometry is sphere or torus, then any value of ncells nx, ny, nz is valid and the code will not use those values.

cellparameters 9.8380486 8.52 12.0 The three numbers after cellparameters are the lengths of the cell vectors **a**, **b** and **c** (also called cell parameters) in angstroms of the simulation cell, respectively. In this example, a=9.8380486, b=8.52 and c=12.0 angstroms. **No default value.**

The values of the cell parameters should be consistent with the geometry keyword. The following values are consistent:

geometry	Values of the cell parameters a b and c
slit pore	a, b, c > 0.0
crystal	a, b, c > 0.0
tube	a and b any value, c > 0.0
sphere	a, b and c any value
torus	a, b and c any value

cellangles 90.0 90.0 90.0 The three numbers after cellangles are the alpha, beta and gamma angles in sexagesimal degrees of the simulation cell, respectively. The angle alpha is the angle between vectors **b** and **c**, the angle beta is the angle between the vectors **a** and **c**, and the angle gamma is the angle between the vectors **a** and **b**. Default values: 90.0, 90.0 and 90.0, respectively.

The values of the cell angles should be consistent with the geometry keyword. The following values are consistent:

geometry	Values of cellangles alpha beta gamma
slitpore	alpha, beta, and gamma /= 0.0
crystal	alpha, beta, and gamma /= 0.0
tube	alpha, beta, and gamma any value
sphere	alpha, beta, and gamma any value
torus	alpha, beta, and gamma any value

combiningrule Good-Hope-Berthelot The combining rule of the LJ potentials. Default value: Good-Hope-Berthelot. Valid values:

Good-Hope-Berthelot $\sigma CH2 = sqrt(\sigma C\ \sigma H2)$

Lorentz-Berthelot $\sigma CH2= (\sigma C + \sigma H2)/2$

$\varepsilon CH2 = sqrt(\varepsilon C\ \varepsilon H2)$ in both combining rules

typeofpotentialsg1 none Type of interaction potential between the surface and one molecule of the first type of gas. This potential include all the interactions of the particles of an infinite surface with one molecule.

If this keyword is not present in the input file or its value is none, then the code uses the interaction potential particle-particle, i.e., typeofpotential1.

Default value: none. Valid values and their meaning:

none No potential surface-gas is used; instead the typeofpotential1 is used

constant The potential surface-gas is constant.

Steele or **steele** The potential surface-gas is the Steele potential. This is valid only for slitpore geometry.

Crowell-Brown or **crowellbrown** The potential surface-gas is the Crowell-Brown potential. This is valid only for slitpore geometry.

Tjatjopoulos or **tjatjopoulos** The potential surface-gas is the Tjatjopoulos potential. This is valid only for tube geometry.

Sideriusgelb or **sideriusgelb** The potential surface-gas is the Siderius-Gelb potential. This is valid only for sphere geometry.

nadditionalsheets 0 Number of additional sheets to be considered in the surface-gas potentials of gas 1 and gas 2. Valid values >= 0. Default value: 0.

symbolgas2 H2 Chemical symbol of the second type of gas of the system. Valid and default values: The same as in the keyword symbolgas1.

typeofpotential2 lj Type of interaction potential of the second type of gas. Valid and default values and their meaning are the same as in the keyword typeofpotential1

typeofpotentialsg2 none Type of interaction potential between the surface and a molecule of the second type of gas. Valid and default values and their meaning are the same as in the keyword typeofpotentialsg1.

potentialenergystep 0.1 To obtain the file potentialenergy.dat, the potential energy is calculated every x angstroms. The value of this keyword is the value of x. A value of 0.001 angstroms can be useful to obtain the potential energy in the regions where the potential energy changes very fast. Valid values > 0.0. Default value: 0.1.

potentialenergymaximum 0.1 The potential energy at point z is printed into the file potentialenergy.dat if it is equal or less than y eV. The value of this keyword is the value of y. Any value is valid. Default value: 0.1.

molarfractiongas1 1.0 Molar fraction of gas 1. Default value: 1.0.

molarfractiongas2 0.0 Molar fraction of gas 2. Default value: 0.0.

The sum of molarfractiongas1 and molarfractiongas2 must be equal to 1.0. If not, the code prints a message and stops the simulation.

If there is only one species of gas, then molarfractiongas1 must be 1.0 and molarfractiongas2 must be 0.0. If not, the code prints a message and stops the simulation.

If there are two species of gases, then molarfractiongas1 and molarfractiongas2 must be larger than zero and less than one. If not, the code prints a message and stops the simulation.

percentagetrialsswapidentity 5.0 On each iteration of a GCMC simulation with two types of gases, there is a trial: movement, addition, deletion or swapping identity trial. This keyword is the probability that the trial will be swapping identity trial. Valid values: 0.0-100.0. Default value: 0.0. Recommended value, if there is a mix of gases: 5.0.

Note: In future versions, I will rename this keyword with a name more self-explanatory and more attached to the meaning of the keyword.

depletion1 0.5 300.15 0.000100 0.100000 0.5 77.00 0.000100 0.100000 There are two groups of numbers. Each group contains four numbers: The depletion pressure in MPa, the depletion temperature in K, the density in kg/L of the stored gas 1 at the depletion pressure and temperature. and the gravimetric capacity in wt. % of the stored gas 1 at the depletion pressure and temperature.

Some keywords are ***mandatory***. Their line keywords should be in the input file: Pressure, Temperature, distance, percentagetrialsmovemolecules, nmoleculesmovedperiteration, nitersbetweenvaluesforaverages, nequilibrationiterations, symbolgas1, nwalls, symbolwall, xwall, ywall, zwall and cellparameters. If the cell

angles are 90, 90, 90 then, it is not mandatory to write the line keyword cellangles and otherwise, it is mandatory.

In the case of ***mix of gases***, there should be also the following line keywords in the input file: molarfractiongas1, molarfractiongas2 and percentagetrialsswapidentity.

3. Output

There are three groups of output files: The file .out, the group of files printed without using keywords, and the files printed using keywords.

3.1 File .out

The main output file usually has the extension out. The name of the file is chosen by the user at the moment of the run.

The next lines are the content of the .out file corresponding to the input example:

```
Run started on Nov 07 2024 02:10:32 by user ivan at node25.cluster09
Version of Sun Nov 10 16:24:10 CET 2024

WARNING: The temperature is equal or below 150 K and quantum corrections
are not applied. It is recommended to apply quantum corrections
at temperatures below 150 K.

WARNING: ncellsa=  1 and it should be at least   3 for cutoff= 20.00 angstroms and this cell.
WARNING: ncellsb=  1 and it should be at least   3 for cutoff= 20.00 angstroms and this cell.

Orthorombic cell
lengths=    9.838048600    8.520000000   12.000000000 angstroms
angles=    90.000000000   90.000000000   90.000000000 degrees
avector=    9.838048600    0.000000000    0.000000000 angstroms
bvector=    0.000000000    8.520000000    0.000000000 angstroms
cvector=    0.000000000    0.000000000   12.000000000 angstroms

Coordinates of the pore are respect to internal coordinates.
Pore center and radius =      0.000     7.000     6.000        6.14 angstrom
Number of pores with radius >= 3.0 angstrom =     1
Average pore radius =                             6.14 angstrom
Length of cubic boxes to calculate pore center and radius = 1.00 angstrom

Volume =              1005.842 angstrom^3
Volume available =       471.910 angstrom^3
Porosity =             46.917 %  =  0.469 dimensionless
```

Volumetric density of atoms = 0.063628 atoms/angstrom^3

Porosity * Volumetric density of atoms = 0.029852 atoms/angstrom^3

Volumetric density of the adsorbent material = 1.268 kg/L

Porosity * Volumetric density of the adsorbent material = 0.594853 kg/L

Surface density of the adsorbent material = 7.607320439E-07 kg/m^2

Specific surface area of the adsorbent material = 2629.047 m^2/g

Mass of the adsorbent material = 0.127529385 10^-23 kg

Surface density of the adsorbent material using H2 = 5.605687239E-07 kg/m^2

Specific surface area of the adsorbent material using H2 = 3567.805 m^2/g

LJ potential

Coefficients of the potential C -H2 : a6= 0.814248 ryd angs^6 a12= 858.026559 ryd angs^12

Coefficients of the potential H2 -H2 : a6= 0.579023 ryd angs^6 a12= 397.405909 ryd angs^12

Coefficients of the potential C -H2 : sigma= 3.190 angstrom epsilon= 0.002628 eV \cite{rlc98}

Coefficients of the potential H2 -H2 : sigma= 2.970 angstrom epsilon= 0.002870 eV \cite{rlc98}

P= 50.00000 MPa T= 77.00 K V= 1005.8 angstrom^3 beta= 2050.485 ryd^-1 lth= 1.401 angstrom mu= -0.000723 ryd/molecule EOS= SRK densityrealgas= 0.071238 kg/L gas= H2

Number of fixed atoms: 64

nmolecules= 32 < Nminstatisticalmeaning= 1000 H2

nestimatedstoredmolecules= 21 < Nminstatisticalmeaning= 1000 H2

nestimatedmoleculesidealgas= 47 < Nminstatisticalmeaning= 1000 H2

nestimatedmoleculesrealgas= 21 < Nminstatisticalmeaning= 1000 H2

Volume = 1006 angstrom^3 < Vminstatisticalmeaning= 24654 angstrom^3 H2

Volume = 1006 angstrom^3 < Vminstatisticalmeaning ideal gas = 21262 angstrom^3 H2

Initial average wall distance: 3.000000 angstroms H2

Estimated computation times:

checkbasis: 0.0 seconds

volumeoccupied: 0.0 seconds

poreradii: 0.0 seconds

write3dpotential: 0.0 seconds

Before MC cycles: 0.0 seconds

MC cycles: 3 seconds

After MC cycles: 0.0 seconds

Total: 3 seconds

it= 0 nmolecules= 32

P= 50.00 MPa T= 77.00 K gc= 5.224205 wt. % nmolecules= 21.000000 densitystoredgas= 0.069888 kg/L gc= 55.121719 mg/g excessgc= 2.795685 wt. % densityexcessgas= 0.036466 kg/L H2 niterations= 5000

P= 50.00 MPa T= 77.00 K gc= 5.224205 wt. % nmolecules= 21.000000 densitystoredgas= 0.069888 kg/L gc= 55.121719 mg/g excessgc= 2.795685 wt. % densityexcessgas= 0.036466 kg/L H2 niterations= 6000

P= 50.00 MPa T= 77.00 K gc= 5.224205 wt. % nmolecules= 21.000000 densitystoredgas= 0.069888 kg/L gc= 55.121719 mg/g excessgc= 2.795685 wt. % densityexcessgas= 0.036466 kg/L H2 niterations= 7000

P= 50.00 MPa T= 77.00 K gc= 5.165114 wt. % nmolecules= 20.750000 densitystoredgas= 0.069056 kg/L gc= 54.465508 mg/g excessgc= 2.733523 wt. % densityexcessgas= 0.035634 kg/L H2 niterations= 8000

P= 50.00 MPa T= 77.00 K gc= 5.129659 wt. % nmolecules= 20.600000 densitystoredgas= 0.068557 kg/L gc= 54.071781 mg/g excessgc= 2.696226 wt. % densityexcessgas= 0.035134 kg/L H2 niterations= 9000

it= 10000 nmolecules= 21

End of the iterations.

E-muN of the last state of the system at 50.00000 MPa 77.00 K: -0.04905767 rydbergs

Number of equilibration iterations according

to the MSER algorithm by White, using:

energy = 3058

mu N = 9575

energy - mu N = 3058

Number of equilibration iterations used in

the simulation = 5000

Average negative potential interaction energy per H2 molecule = -0.019703 eV, using a step of 2.000 angstroms

	Actual Percentage trials/Niterations	Input Percentage trials/Niterations	Naccepted	Ntrials	Percentage trials/accepted
Move	20.360	20.000	1658	2036	81.434
Add	39.420	40.000	27	3942	0.685
Remove	40.220	40.000	38	4022	0.945

P= 50.00 MPa T= 77.00 K gc= 5.145417 wt. % nmolecules= 20.666667 densitystoredgas= 0.068779 kg/L gc= 54.246771 mg/g excessgc= 2.712802 wt. % densityexcessgas= 0.035356 kg/L usabledensity= 0.068679 kg/L usablegc= 5.055267 wt. % autonomy05= 862.1 km H2 niterations= 10000

Volume of the deposits= 122.4 L. Fuel consumption= 0.00975075 kg/km.

Average nearest-neighbour H2-H2 distance at 77.00 K: 3.022087 angstroms

P= 50.00 MPa T= 77.00 K isosteric heat of adsorption= 0.00361736 rydbergs = 0.04921668 eV = 4.74868779 kJ/mol H2

Average wall distance at 77.00 K: 4.006756 angstroms

sigma_<A>= 0.016705 R=sigma_<A>/average= 0.004169 FOM=1/(R^2 T)= 5.E+04 H2

nestimatedstoredmoleculesfs= 86.1 H2

nestimatedstoredmolecules= 21.3 H2

nestimatedmoleculesidealgas= 47.3 H2

nestimatedmoleculesrealgas= 21.4 H2

Real time per interaction: 1.5E-07 seconds

Estimated time per interaction: 3.2E-07 seconds

The sum of the real number of interactions is 7536122

The sum of the estimated number of interactions is 9979055

Absolute difference between the sums of the

estimated and real number of interactions = 2442933

Relative difference between the sum of the

estimated and real number of interactions = 32.4 %

Computation times:

Before MC cycles: 0.1 seconds

MC cycles: 1 seconds

After MC cycles: 0.0 seconds

Total time: 1.3 seconds at node25.cluster09

Run ended on Nov 07 2024 02:10:33 by user ivan at node25.cluster09

The ***first line*** of the output file shows when the simulation was started and by which user and at which computer. The ***second line*** of the output file indicates the version of the mcmgs code:

Run started on Nov 07 2024 02:10:32 by user ivan at node25.cluster09

Version of Sun Nov 10 16:24:10 CET 2024

The versions of the code are labelled according to the date of the version. The versions are not labelled following some alphabetic and/or numerical order. In the present example, it is the version of November 10 2024 at 16:24.

The next lines could be warnings or the results of some calculations. After those lines, the code prints the results at some iterations. For instance, the line

it= 0 nmolecules= 32

means that at the beginning of the simulation, iteration 0, the number of molecules was 32.

Other results printed at some iterations are:

P= 50.00 MPa T= 77.00 K gc= 5.224205 wt. % nmolecules= 21.000000 densitystoredgas= 0.069888 kg/L gc= 55.121719 mg/g excessgc= 2.795685 wt. % densityexcessgas= 0.036466 kg/L H2 niterations= 5000

P= 50.00 MPa T= 77.00 K gc= 5.224205 wt. % nmolecules= 21.000000 densitystoredgas= 0.069888 kg/L gc= 55.121719 mg/g excessgc= 2.795685 wt. % densityexcessgas= 0.036466 kg/L H2 niterations= 6000

P= 50.00 MPa T= 77.00 K gc= 5.224205 wt. % nmolecules= 21.000000 densitystoredgas= 0.069888 kg/L gc= 55.121719 mg/g excessgc= 2.795685 wt. % densityexcessgas= 0.036466 kg/L H2 niterations= 7000

P= 50.00 MPa T= 77.00 K gc= 5.165114 wt. % nmolecules= 20.750000 densitystoredgas= 0.069056 kg/L gc= 54.465508 mg/g excessgc= 2.733523 wt. % densityexcessgas= 0.035634 kg/L H2 niterations= 8000

P= 50.00 MPa T= 77.00 K gc= 5.129659 wt. % nmolecules= 20.600000 densitystoredgas= 0.068557 kg/L gc= 54.071781 mg/g excessgc= 2.696226 wt. % densityexcessgas= 0.035134 kg/L H2 niterations= 9000

These lines contain the ***gravimetric and volumetric storage capacities*** of hydrogen (or the corresponding gas) of the porous solid material at a pressure, a temperature and an iteration. The code prints this information every nprintaverage iterations, after nequilibrationiterations. nprintaverage=min(maxniterations/10,1000000). In our example, maxniterations=10000, nequilibrationiterations=5000 and nprintaverage=1000. Hence, we see this type of lines at the iterations 5000, 6000, 7000, 8000, 9000 and 10000.

After the storage capacities, we find the lines:

it= 10000 nmolecules= 21

End of the iterations.

These lines mean that the number of molecules in the iteration 10000 was 21 and that this was the last iteration. Then, we find the lines:

	Actual Percentage trials/Niterations	Input Percentage trials/Niterations	Naccepted	Ntrials	Percentage trials/accepted
Move	20.360	20.000	1658	2036	81.434
Add	39.420	40.000	27	3942	0.685
Remove	40.220	40.000	38	4022	0.945

These lines indicate the number of trials accepted on each type of trial and the corresponding percentage. There are many trials whose result was not accepted.

Then, we reach the ***most important line*** of the .out file:

P= 50.00 MPa T= 77.00 K gc= 5.145417 wt. % nmolecules= 20.666667 densitystoredgas= 0.068779 kg/L gc= 54.246771 mg/g excessgc= 2.712802 wt. % densityexcessgas= 0.035356 kg/L usabledensity= 0.068679 kg/L usablegc= 5.055267 wt. % autonomy05= 862.1 km H2 niterations= 10000

This is the ***last line containing the storage capacities*** and is the relevant one, because it contains the values of the ***storage capacities at the end of the simulation.***

The ***penultimate line*** of the .out file contains the computation time of the simulation and the name of the computer that ran the simulation. The ***last line*** of the .out file shows when the simulation finished and by which user at which computer:

Total time: 1.3 seconds at node25.cluster09

Run ended on Nov 07 2024 02:10:33 by user ivan at node25.cluster09

3.2 Files printed without using keywords

The files energies.dat, sigma2vsiteration.dat, histogrammolecules.dat and safety.xyz are printed without using the keywords.

If there are two types or species of gas, then the code prints also the file histogrammoleculesgas2.dat.

The files energies.dat, sigma2vsiteration.dat and histogrammolecules.dat (and also histogrammoleculesgas2.dat, if there is a second type of gas) are always printed. The file safety.xyz is rewritten ***every 12 hours.*** If the run takes less than 12 hours, then this file is not printed.

The formats of the file energies.dat, sigma2vsiteration.dat and histogrammolecules.dat are ready for the ***xmgrace*** code.

energies.dat: This file contains the following columns: iteration, energy in Rdybergs of the system, the product μN, where m is the chemical potential and N is the number of molecules, the energy – μN in Rydbergs and finally the average distance between the molecules in angstrom.

An example of the content of the file energies.dat:

```
# Iteration Energy(Rydberg) mu*N(Rydberg) Energy-muN(Rydberg) d(A)
      0       0.58792012   -0.02314549       0.61106560   2.887510
    500       -0.03364078   -0.01084945       -0.02279133   3.886445
…….
   9500       -0.06594599   -0.01591252       -0.05003347   3.386689
  10000       -0.06216896   -0.01518923       -0.04697973   3.371292
```

The first line is the title and meaning of the columns below. The next lines are the results of the simulation, sorted by columns.

sigma2vsiteration.dat: This file contains the following columns: Iteration, energy in Rydbergs, average energy in Rydbergs and ^{2}. The first line is the title and meaning of each column. The next lines are the results of the simulations. The code prints these data at the iterations 10000, 10000+nprint, 10000 + 2nprint, etc.

An example of the content of the file sigma2vsiteration.dat:

```
# Iteration Energy-muN  Average     Sigma^2
#                   Energy-muN
#          Rydberg    Rydberg
    10000   -0.04698   -0.04078    0.000034019
```

histogrammolecules.dat: This file contains a histogram of the molecules of the first type of gas inside the porous material, in the last iteration of the simulation.

This file contains the following columns: Distance in angstroms to one border of the porous solid material, position in angstroms with respect to the center of the porous solid material and number of molecules. The structure is such that when this file is plotted using xmgrace, the results are a group of columns, i.e., a histogram.

An example of the content of the file histogrammolecules.dat:

```
# Position    Distance  Number of
# respect     to left   molecules
# pore        surface   of H2
# center      (angstrom)
# (angstrom)
   -6.000     0.000     0
   -5.900     0.100     0
   -5.900     0.100     0
……..
    6.000    12.000     0
    6.000    12.000     0
    6.100    12.100     0
```

histogrammoleculesgas2.dat: Idem as histogrammolecules.dat, but of the second type of gas.

safety.xyz: It has the format of an xyz input file of the mcmgs code. This file contains all the data to restart a calculation: The first part (symbols, coordinates, and states of the particles) and the second part (the keyword lines). The purpose of this file is to restart a calculation in case the run was too long, and something stopped it.

3.3 Files printed using keywords

intermediategeometries.xyz: This file contains the geometries (symbols, coordinates, and states of the particles) of intermediate geometries. For instance, the geometries at the iterations 500, 1000, 1500, 2000, etc. This file is usually very large. The purpose of this file is to use it to make a movie of the simulation.

lastgeometry.xyz: This file contains the geometry (symbols, coordinates, and states of the particles) of the last iteration of the simulation, plus the keyword lines used. This file has the format of an .xyz input file of the mcmgs code. Hence, this file can be used to continue the simulation.

potentialenergy.dat: This file contains the potential energy in Rydberg of a molecule of the first type of gas interacting with the solid porous material vs some geometric parameter of the solid porous material. The content of this file is independent of the Pressure, Temperature, and other parameters. It depends on the symbols and coordinates of the particles, the type of geometry and the type of gas.

This file contains the following columns: The z coordinate of a gas molecule in angstroms, inside the porous material, the z-center of the pore position of the gas molecule in angstrom, the energy in rydbergs of one gas molecule interacting with the porous material and the energy in eV.

An example of the content of the potentialenergy.dat file:

```
#    z       z-center        Energy          Energy
# (angstrom)  (angstrom)      (Rydberg)       (eV)
   2.600000   -3.400000     0.005265643     0.071642715
   2.700000   -3.300000     0.001547612     0.021056329
…….
```

The first line is the title and meaning of each column. The next lines are the results of the calculations. The format of potentialenergy.dat is ready for the code xmgrace.

potentialenergygas2.dat: Idem as potentialenergy.dat, but the potential energy is of the second type of gas.

4. How to run the code

4.1 Before running the code

The files related to the Monte Carlo simulations and the mcmgs code are in the directory mcmgs. To move or change to the mcmgs directory, enter:

```
cd $HOME/mcmgs
```

There are three sub-directories under the mcmgs directory: source, tests and scripts.

Create a new sub-directory. You will make simulations in the new sub-directory. Enter the following commands:

```
mkdir name-of-new-sub-directory
cd name-of-new-sub-directory
ln -s ../scripts/mcmgs.sh mcmgs.sh
```

Then, you can create new input files and start new simulations.

4.2 Directly

The mcmgs code is executed directly, without using scripts, as follows. At the Linux prompt, enter:

```
mcmgs < name.xyz > name.out &
```

The code prints into the *default output* (the screen). The operation '> name.out' *redirects the default output* to the file name.out.

The name and extension of the output file could be any, but we *recommend* using a name that is self-explanatory and to use the extension out

The name and extension of the input file also could be any, but again, we *recommend* using a self-explanatory name and to use the extension xyz.

4.3 Using the script mcmgs.sh

The script mcmgs.sh contains many jobs. Each job consists of two main tasks:

a) First, the script *writes a mcmgs input file*, starting from a *generic xyz file.*

A generic xyz file is a file with the correct structure and data of an mcmgs input file, but with generic values for the keywords. For instance, a generic file contains lines like the following:

distance distancevar

ensemble mctype

eos eosvar

The script mcmgs.sh replace mctype by a name chosen by the user, for instance, by muvt. And so on with distancevar, eosvar, etc.

A generic xyz file is very useful, because this same file can be used for different types of simulations: Different pressures and/or temperatures, etc.

b) The *second task* performed by the script mcmgs.sh is to run the Monte Carlo simulation.

To run the script and the mcmgs code, at the Linux prompt, enter:

cd $HOME/mcmgs/system

and then enter

$HOME/mcmgs/system/mcmgs.sh run xyzfile job &

There are three fields after the name of the script, mcmgs.sh. The name system could be mof5, hkust1, nanoonions, etc.

The ***first field*** is the ***operation.*** In this example, the operation is run, and it means to run the simulation.

The ***second field*** is the name of the ***generic xyz file, without the xyz extension.*** The scripts looks for a file with that name, without the xyz extension, in the present directory.

The ***third field*** is the name of the ***job.*** A job, as we have explained, consists on a set of values of the keywords: Pressure, Temperature, distance, ensemble, etc. The mcmgs.sh scripts contains many jobs and their corresponding set of values of the keywords.

For instance, the job ‘muvtperiodic110’ means:

ensemble muvt

boundaryconditions periodic

ncells 1 1 0

The job P1T80.15periodic110 means:

Pressure 1

Temperature 80.15

boundaryconditions periodic

ncells 1 1 0

The job P1T80.15periodic110it1000000 means:

Pressure 1

Temperature 80.15

boundaryconditions periodic

ncells 1 1 0

maxniterations 1000000

nequilibrationiterations 500000

The job P1T80.15periodic110it1000000fh means:

Pressure 1

Temperature 80.15

boundaryconditions periodic

ncells 1 1 0

maxniterations 1000000

nequilibrationiterations 500000

quantumcorrections yes

The script mcmgs.sh creates a temporary directory under mcmgs/system and writes the output files in that temporary directory. In this example, it writes the file in the directory $HOME/mcmgs/system/tmp-$xyzfile-$job

Example:

$HOME/mcmgs/system/mcmgs.sh run slitpore-w07.0 P1T80.15periodic110 &

To move the output files from the temporary sub-directory to mcmgs/system and to remove the temporary sub-directory, at the Linux prompt, enter:

cd $HOME/mcmgs/system

and then, enter

$HOME/mcmgs/system/mcmgs.sh move xyzfile job

Example:

$HOME/mcmgs/system/mcmgs.sh move slitpore-w07.0 P1T80.15periodic110

4.4 Using the script sendjobstocluster.sh

The script sendjobstocluster.sh sends several jobs (tens or hundreds of jobs) to the cluster. A job can be a mcmgs simulation, a calculation carried out with another code, another script, etc.

4.4.1 How to run the script sendjobstocluster.sh

To run this script and the mcmgs code, follow these two steps:

a) Write the file pendingjobs.dat

This file should contain all the jobs that you want to run. Each line of that file is a job. The number of pending jobs is *npjobs*.

A job can consist of only one GCMC simulation or of a group of GCMC simulations.

An example of the content of pendingjobs.dat:

$HOME/mcmgs/fullerene/mcmgs.sh run c720withh2 P0.1T298.15

$HOME/mcmgs/fullerene/mcmgs.sh run c720withh2 P0.5T298.15

$HOME/mcmgs/fullerene/mcmgs.sh run c720withh2 P1T298.15

$HOME/mcmgs/fullerene/mcmgs.sh run c720withh2 P2T298.15

……

……

b) Run the script:

At the Linux prompt and in the directory that contains pendingjobs.dat and the .xyz files, enter:

sendjobstocluster.sh period ndays &

The variable *period* is in seconds. The script will be running during *ndays* at most. The script will be running during nc cycles, where nc = *ndays * 86400 / period*.

Example:

sendjobstocluster.sh 1000 90 &

4.4.2 How the script sendjobstocluster.sh works

a) The script *scans the cluster* to find out which nodes are available and how many processors are available and write the results in availablenodes.dat. The script deletes and rewrites availablenodes.dat at the beginning of every cycle.

The number of available processors after the scan is *nap*. There are *nap* lines in the file availablenodes.dat. Each line specifies an available processor in an specific node of the cluster. If there are k processors available in the node compute-0-x, then there will be k lines about the node compute-0-x in the file availablenodes.dat

An example of the content of the file availablenodes.dat:

compute-0-2

compute-0-3

compute-0-8

compute-0-9

compute-0-12

compute-0-14

compute-0-19

compute-0-22

compute-0-23

compute-0-24

compute-0-25

compute-0-26

compute-0-27

compute-0-29

compute-0-31

compute-0-2

compute-0-3

compute-0-8

compute-0-9

…...

b) The script *sends the first n jobs* written in the file pendingjobs.dat to the available processors of the cluster, where *n* is the minimum of nap and npjobs.

If *nap* is smaller than *npjobs*, then *n=nap* and the script will send the first *n* jobs and the rest of the jobs, *npjobs-n*, will wait to be sent in the next cycle or cycles.

If *nap* is equal or larger than *npjobs*, then *n=npjobs* and all the pending jobs will be sent and the file pendingjobs.dat will be empty.

c) *Remove the n sent jobs* from the file pendingjobs.dat. The number of lines of pendingjobs.dat decreases as the script runs. The new number of pending jobs will be *npjobs = npjobs – n***.**

d) *Wait 'period'* seconds. In the example of section 4.4.1, period=1000.

e) *Stop* after *'ndays'* days or if the file pendingjobs.dat is empty. When all the jobs are sent, the file pendingjobs.dat is empty. In the example of section 4.4.1, *ndays*=90.

f) *Repeat* the tasks a)-d). A cycle is composed by the tasks a)-d).

4.5 How to sort the temporary files and subdirectories

The temporary subdirectories contain the output files. The output files have the suffix -new. The script mcmgscheckcalculations.sh renames and moves the output files to the main directory. It also removes the empty subdirectories, after issuing a message in the screen.

If the simulation corresponding to a temporary sub-directory has not finished yet, then the script issues a warning and does not move the files, nor the sub-directory.

To run the script, enter:

mcmgscheckcalculations.sh

4.6 How to backup the files

It is convenient and a security measure to backup the content of the sub-directory where you are making the simulations. ***To backup the sub-directory,*** enter:

cd

tar cvf mcmgs-nameofsub-directory.tar mcmgs/nameofsub-directory

gzip mcmgs-nameofsub-directory.tar

mv -f mcmgs-nameofsub-directory.tar.gz backup/mcmgs/.

The backup file mcmgs-nameofsub-directory.tar.gz contains the sub-directory and the files of that sub-directory. If you delete accidentally a file, then you can retrieve from the backup file.

To ***retrieve the file*** name1.xyz, enter:

cd

mkdir tmp

cd tmp

tar zxf ../backup/mcmgs/mcmgs-nameofsub-directory.tar.gz

The file name1.xyz will be in $HOME/tmp/mcmgs/nameofsub-directory/name1.xyz

To ***move this file*** to your sub-directory, enter:

cd

mv tmp/mcmgs/nameofsub-directory/name1.xyz
mcmgs/nameofsub-directory/name1.xyz

or

mv $HOME/tmp/mcmgs/nameofsub-directory/name1.xyz $HOME/mcmgs/nameofsub-directory/name1.xyz