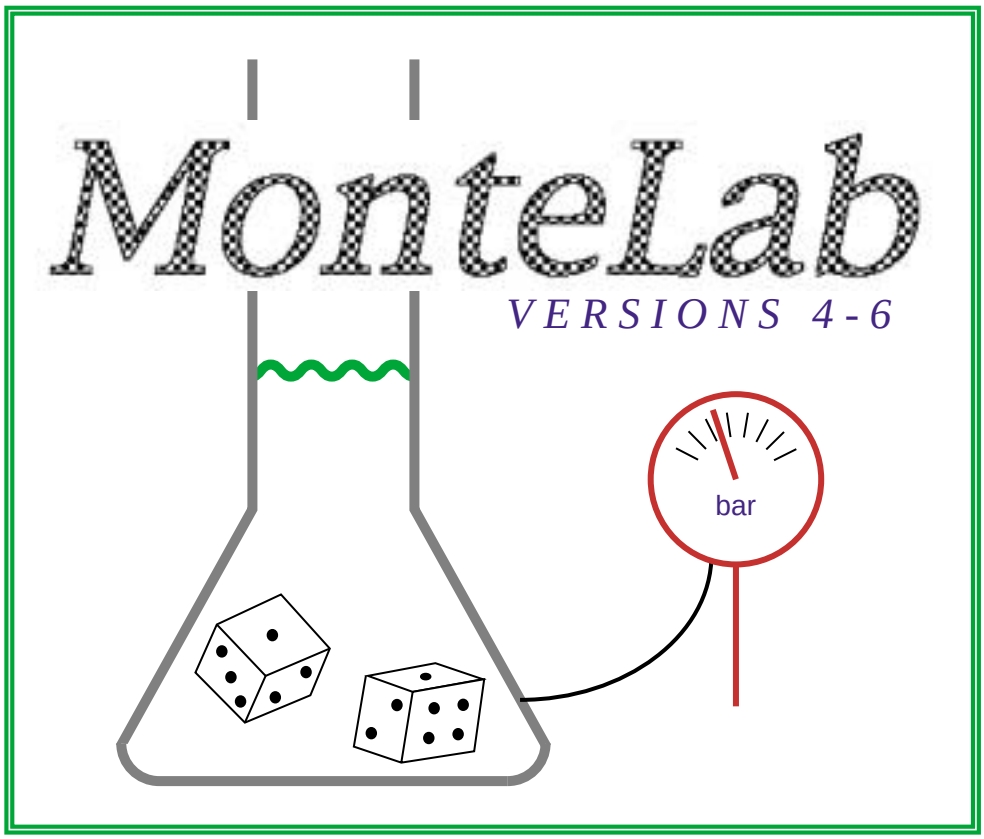

Programmer's Guide

Jan Westergren Sture Nordholm
Department of Experimental Physics - Chalmers University of Technology
Department of Physical Chemistry - Göteborg University
Göteborg - Sweden

April 2005

About MonteLab

MonteLab is a program that employs the Metropolis Monte Carlo simulation method for calculating thermodynamical properties in bulk or particle systems (clusters). It was originally developed as an educational program under the VisAb project ("A Visualization of Abstract Concepts in Physical Chemistry by Computer" [1]) which was financed by Högskoleverket (the Swedish National Agency for Higher Education) but it has now also found its way into scientific research. MonteLab is now introduced as free software and might be downloaded from the MonteLab webpage *www.phc.gu.se/~janw/MonteLab.htm*. The program is written in standard Fortran and should be compilable for Fortran 77 and later version. Bugs reports and improvement suggestions are very welcome to Sture.Nordholm@phc.gu.se.

We believe that MonteLab can serve as a Monte Carlo tool for many people. A necessity when distributing software to new users is a good and extensive documentation. Hopefully, with the User's Guide and with a well-commented code, MonteLab will be easy to use and understand. The User's Guide is downloadable from the MonteLab webpage.

The number of possible areas of use for a Monte Carlo program is of course enormous, and a program that is not easy to modify has no future outside the original users. Therefore, much effort has been put into writing a clear and organized code with many comments. The program is organized as a main program that should suit many different systems and modules that may be modified for different systems. The Programmer's Guide is intended to give understanding of the code.

This version of the User's Guide covers the following systems:

1. Simple fluids with Lennard-Jones 12-6 (LJ) or Morse pairwise interaction.
   The canonical ensemble (NVT).
2. Simple fluids with cell constraints with LJ or Morse potential.
   The canonical ensemble (NVT).
3. Clusters of atoms interacting with the MBA (Many-Body Alloy) potential.
   The canonical ensemble (NVT).
4. Clusters of atoms interacting with the MBA (Many-Body Alloy) potential.
   The microcanonical ensemble (NVE).
5. Clusters of atoms interacting with the MBA (Many-Body Alloy) potential
   with a reference system. The microcanonical ensemble (NVE).

The policy of version number is that the same input data (as e.g. the initial random number) should always generate exactly the same output (as long as the same computer is used) when the same version of the program has been used.

Jan Westergren
April 2005

MonteLab version 6
www.phc.gu.se/~janw/MonteLab.htm

Jan Westergren
Department of Experimental Physics
Molecular Physics Group
Chalmers University of Technology and
Göteborg University
Göteborg - Sweden

Professor Sture Nordholm
Department of Physical Chemistry
Göteborg University
Göteborg - Sweden

Any contact about MonteLab should be taken with Professor Sture Nordholm
e-mail Sture.Nordholm@phc.gu.se

Every reasonable effort has been made to eliminate errors in the program, but the authors cannot assume responsibility for the consequences of their use.

GENERAL INFORMATION FOR ALL MONTELAB PROGRAMS

THE MAIN PROGRAM

The calls for modules

MonteLab consists of a main program and a number of modules. The main program is fairly general and should suit many systems. Each module is an own file which is called by an `include` statement. Each module consists of a number of subroutines and functions that are required for that module. Some functions, as e.g. energy calculation, is called from different modules but is of course only defined in one of them, in most cases where it first appears. Local variables are used sparingly. The global variables are all defined in one file, COMMON_Global_variable_declaration.Mo6. The suffix Mo6 indicates version number.

One trial move of a particle/atom is called one *Monte Carlo step*.

The main program consists of the following module calls.

1. General_info

Brief on-line information

2. Define_system

The user is asked to enter parameters as number of particles, volume and temperature.

3. Define_potential

Input of parameters related to the potential model.

4. Initial_configuration

The initial configuration is generated, either by MonteLab or read from an input file.

5. Open_output_files

Open the appropriate output files.

6. Initialize_sums

All quantities as e.g. the potential energy are calculated for the initial configuration. The sum of e.g. the potential energy of all configurations is initiated.

7. Ask_meltdown()
Module asking whether meltdown is to be used.

(8. Ask_steplength_adjustment()
Module asking whether the steplength should be adjusted to give a desired rejection rate. The adjustment can only be done during the meltdown and thus the question is only asked if meltdown is chosen.)

9. Input_steplength
Input of steplength of the moves.

10. Input of a positive integer seed random number.
This question has not got an own module.

(11. Setup_meltdown_execution
Only if meltdown is chosen. Here the number of steps for the meltdown is entered.)

(12. Meltdown_and_adjusting *or* Meltdown_without_adjusting
If meltdown is chosen, the simulation is started from one of these modules. Note that the module that actually runs the simulation (Simulation_loop) is called from these modules.)

(13. Initialize_sums
After the meltdown, all quantities are reset and the final configuration of the meltdown simulation is taken as the initial one in the real simulation.)

(14. Mark_initialization_in_files
In all the output files, the end of the meltdown is written.)

15. Setup_real_simulation_execution
Here the number of steps for the real simulation is entered.

16. Real_simulation_execution
This module starts the real simulation (Simulation_loop)

17. Write_final_output
All results are written to files here.

18. Close_output_files
Close the output files.

Modules that are not called from the main program:

Simulation_loop
Taking care of the Metropolis Monte Carlo loop.

Write_output
Writing intermediate output

Variable names

The variable names are intended to be descriptive with a prefix that describes the kind of variable.

FUN	function
RANDOM	random number generator
SYS	variables defining the system
SAMP	sampling variables
PRES	present value of observables
SUM	accumulated values of observables
x_coord	spatial coordinates
PARAM	potential parameters
Steps	giving number of MC steps
IND	indicators
Answer	answer of questions
NAME	names
CELL	cell limits
CONSTANT	physical constants
PROPOSED	observables proposed before it is clear if the move is accepted
ADJACENT	in cluster simulations: adjacent matrix
NOW	local means
LOC	local variables
TEMP	temporary (local) variables

Averages of quantities as e.g. the potential energy is calculated by accumulating the values for the configuration in SUM variables. Variables with the suffix tail are with tail contribution included. Variables with the suffix sq means the quantity squared. Sums with the suffix LAST are the sum of a quantity the last time the quantity was printed.

Write_output

By a local average we mean the average of a quantity since the last Monte Carlo step for which the quantity was written to the file. With the total average we mean the average since the start of the simulation. Note that the total average is reset after the meltdown. The local averages are marked by NOW.

Meltdown_and_adjusting

The outer loop is over the ten different steplengths. The inner loop is over the simulation loop. After each round of the inner loop, the pressure is evaluated and quantities are written to the files.

SIMPLE FLUIDS WITH LENNARD-JONES 12-6 (LJ) OR MORSE PAIR-POTENTIALS IN THE CANONICAL ENSEMBLE (NVT)

`Initial_configuration`

`SUBROUTINE: Fetch_initial_configuration`

The reading from the configuration input file works as follows: The input file is read through and the number of rows with an integer in the first column (`LOC_Integerrows`) is counted. The first column in the last line should equal the number of particles, if not the reading of the file failed. If it is correct, the file is rewound and the ending lines are read. The coordinates are checked to be within the cube.

`Simulation_loop`

For a system with a pair potential, the change in energy after the trial move is the difference in removal energy for the active particle. Thus the simulation loop goes through the following parts.

1. Choose particle to be moved.
2. Calculate removal energy for that particle before and after the trial move.
3. Calculate the logarithm of the difference in Boltzmann factor and use a random number to determine if the trial move shall be accepted.
4. Update the present potential energy and calculate also the present values of the other quantites.
5. Update the accumulated sums of the quantites.

The simulation module actually consists of two parallel loops. One is for the LJ potential and one for the Morse potential.

SIMPLE FLUIDS WITH CELL CONSTRAINTS WITH LJ OR MORSE POTENTIAL IN THE CANONICAL ENSEMBLE (NVT)

The program is almost identical to MonteLab FLUPAIR (Chapter 1)

CLUSTERS OF ATOMS INTERACTING WITH THE MBA (MANY-BODY ALLOY) POTENTIAL. THE CANONICAL ENSEMBLE (NVT)

The matrix `ADJACENT_atoms` is the adjacent matrix.

`PRES_G_sampling` and `SUM_G_sampling` are used originally for the $g(r)$ function but the same variables are used for the $s(r)$ sampling. $s(r)$ might be called $g(r)$ in the code since the suvroutines concerning $g(r)$ is still used for $s(r)$.

`PRES_Distance` the distance between atoms i and j.

`SUBROUTINE: Calculate_radii_of_gyration`

The radii of gyration is calculated from the moments of interia along the principal axes. These moments of interia are the eigenvalues of the matrix **I**. To obtain the eigenvalues a third degree equation has to be solved.

The potential energy

The potential energy is calculated in terms of

$$\text{Pot\_B} = \sum_{i,\, j \neq i}^{N} \varepsilon_0 \exp\left(-p\left(r_{ij}/r_0 - 1\right)\right)$$

$$\text{Pot\_B\_surf} = \sum_{\substack{i,\, j \neq i \\ i \in \text{surface atoms}}} \varepsilon_0 \exp\left(-p\left(r_{ij}/r_0 - 1\right)\right)$$

$$\text{Pot\_A2}(i) = \sum_{j \neq i}^{N} \xi_0^2 \exp\left(-2q\left(r_{ij}/r_0 - 1\right)\right)$$

$$\text{Pot\_A1} = -\sum_{i} \sqrt{\text{Pot\_A2}(i)}$$

$$\text{Pot\_A1\_surf} = -\sum_{\substack{i \in \text{surface} \\ \text{atoms}}} \sqrt{\text{Pot\_A2}(i)}$$

MonteLab either goes to the subroutine `Simulation_loop_Surface` or to the `Simulation_loop`, depending on the user's choice. Since the simulation loop is rather CPU-consuming, priority is given to an optimized code.

The simulation loop works as follows.

The trial increase in energy is calculated. To avoid recalculating the variables Pot_A2(i) if the trial move is accepted, the trial values are temparaily stored as PROPOSED varialbles. The same is for the interatomic distances and the adjacent matrix. If the trial moce is accepted the PROPOSED values are transferred to the present variables (PRES).

If the surface energy shall be calculated, the energies are calculated in two loops. If the trial atom is a surface atom, loop number one is over all other surface atoms and loop number two is over the core atoms. If the trial atom is a core atom, the first loop is over the surface atoms and the second loop is over all other core atoms.

SIMPLE FLUIDS WITH CELL CONSTRAINTS WITH LJ OR MORSE POTENTIAL IN THE MICROCANONICAL ENSEMBLE (NVE)

The program is almost identical to MonteLab MBACLUSTER (Chapter 3)

SIMPLE FLUIDS WITH CELL CONSTRAINTS WITH LJ OR MORSE POTENTIAL IN THE CANONICAL ENSEMBLE (NVT), ANNEALING

The program is almost identical to MonteLab MBACLUSTER (Chapter 3). The difference is that the steplength might be adjusted during the real simulation as well. The controll of this procedure is taken care of the routines Real_with_adjusting *and* Real_without_adjusting.