NATIONAL TECHNICAL UNIVERSITY OF ATHENS
INTERDEPARTMENTAL GRADUATE PROGRAMMES: MATHEMATICAL MODELLING IN MODERN TECHNOLOGIES AND FINANCIAL ENGINEERING, MICROSYSTEMS AND NANODEVICES, COMPUTATIONAL MECHANICS, MATERIALS SCIENCE AND TECHNOLOGY

MOLECULAR SIMULATIONS OF MATERIALS
Spring Semester 2026

Computer Problems

Due: Monday, June 22nd, 2026

You are asked to solve one of the attached molecular simulation problems.

Each student will submit one report. The report should discuss and explain the problem formulation, the flow of calculations (logical diagram), the computations conducted, the results obtained (tables diagrams, estimation of errors), and the conclusions derived.

FORTRAN codes for the solution of the problems and example input files can be found within the directories molsim_CH4_MD_2026 and molsim_Gibbs_MC_2026, which are available in compressed form through the following link:

http://comse.chemeng.ntua.gr/data/molsim2026.zip

password: m0ls1m2026 (the second character is zero – the fifth is one)

You can solve the problem of your choice on any computer platform available to you. Installation and use of the Intel Fortran Compiler, which is freely available to NTUA students under the Windows and Linux operating systems, is recommended. Installing this compiler presupposes that the Microsoft Visual Studio computational environment has already been installed on your system. This environment is also provided free to NTUA students. Instructions for downloading and installing Microsoft Visual Studio and the Intel Fortran Compiler (Intel Parallel Studio XE), along with details concerning the creation, compilation, and execution of a Fortran program are available in the file Visual_Studio_Fortran_Installation_Guide.pdf, which is accessible through the link provided above.

Computer Problems on Molecular Simulations

FORTRAN codes for the following two problems have been made available to you as explained on the first page of the problem statement. You can transfer the codes to the machine on which you wish to work.

Problem I: Molecular Dynamics of Liquid Methane

A potential parameter set that can reproduce the liquid-phase properties of methane reasonably well has been proposed by Murad and Gubbins [Murad, S., Gubbins, K.E. (1978). Molecular dynamics simulation of methane using a singularity-free algorithm. In *Computer Modeling of Matter*, Lykos, P. (ed.), ACS Symp. Ser. 86, Washington, D.C.]. The molecule is represented as a rigid polyatomic consisting of one carbon and four hydrogen Lennard-Jones centers. The following interaction parameters are used:

	ϵ_{ij}/k_B (K)			σ_{ij}(Å)	
	C	H		C	H
C	51.198	23.798	C	3.350	2.995
H	23.798	8.631	H	2.995	2.813

In subdirectory $CH4$ of the above workstation account is contained code for performing NVE molecular dynamics simulations of a fluid phase consisting of such model methane molecules. The code employs the constraint dynamics algorithm of Edberg, Evans, and Morris, which we discussed briefly in class. The author of the code, Randy Q. Snurr, has included some directions on its use in a README file. It is suggested that you look at this file before anything else.

You are asked to use this MD code to simulate methane under the following conditions:

Temperature $T = 185\text{K}$

Mass density $\rho = 251.3\text{kg/m}^3$

From your simulation estimate:

a. The heat of vaporization of methane at 185K.

b. The equilibrium pressure of the liquid methane at the above temperature and density.

c. The partial pressure of methane gas that would be in equilibrium with the above liquid phase.

d. The self-diffusivity of methane molecules in the liquid under the above conditions.

e. The molecular velocity autocorrelation function and its Fourier transform.

f. The center-of-mass pair distribution function $g(r)$ under the above conditions.

The code you will need to accomplish tasks (a) to (e) above is in place. You may wish to introduce some modifications to address the specific questions we pose. You will have to write *your own* subroutine for calculating the pair distribution function.

Makefiles that you can use to generate the executables from the FORTRAN code are available in subdirectory $CH4$. Two kinds of codes are provided: those used for the molecular dynamics simulation per se (which you need not modify), and those used for postprocessing (which you may need to augment or modify somewhat).

Brief list of the routines used for generating MD trajectories (Subdirectory Src)

mddriver.F : Driver MD routine, which controls all calculations.

evans.F : Calculates constraint forces for use in the Edberg-Evans- Morris algorithm.

gear.F : Numerical integration of the equations of motion with Gear method.

getf.F : Controls calculation of intermolecular forces.

gqbfgs.F : BFGS optimization routine used to restore molecular geometry, which may be distorted somewhat during the numerical integration of the equations of motion.

histo.F : prints out the distribution of molecular energies.

input.F : input of molecular geometry and energetics.

intconf.F : generates an initial configuration for the simulation by placing molecules on an *fcc* lattice and randomizing them with a short Monte Carlo run.

makemol.F : constructs a methane molecule at given orientation.

matslve.F : linear matrix equation solver.

mbvdist.F : random generation of a Maxwell-Boltzmann distribution of velocities.

mkcnstv.F : formation of the constraint vectors, dictated by the rigid molecular geometry.

monte.F : Monte Carlo routine to relax an *fcc* lattice of methane molecules.

mxm1.F : Auxiliary routine for matrix-matrix multiplication

mxv1.F : Auxiliary routine for matrix-vector multiplication

pairfce.F : Loop through all pairs of molecules to determine pair interactions.

pbc.F : Imposes periodic boundary conditions whenever a molecule exits the simulation box.

penalty.F : Controls the optimizations to restore molecular shape.

phi.F : Bond length penalty functions used in molecular shape restoration.

placeh.F : Auxiliary routine for placing hydrogens.

psi.F : Velocity penalty function used for restoration of molecular shape.

rescale.F : Rescales molecular velocities to restore "molecular" temperature (*i.e.*, temperature defined in terms of molecule center-of-mass motion).

setup.F : Initialization and setting up data.

tcalc.F : calculation of "molecular" and "atomic" temperatures.

wtime.F : time information and error handling.

Brief list of post-processing routines for pressure and chemical potential (subdirectory Press)

press.F : Main program controlling Widom test particle insertion for chemical potential and virial theorem calculations of pressure.

finish.F : Final caculations for μ^{ex} (You may want to modify this!)

histo.F : Generates distribution of molecule energies.

input.F : Input of system parameters.

makemol.F : generation of Cartesian coordinates for all atoms on a methane molecule, given the center of mass coordinates and the Eulerian angles.

pair.F : Calculation of test particle energy.

pairmg.F Version of pair.F that uses a truncated and shifted LJ potential.

setup.F : Initialization of arrays, setting up data.

tempcalc.F : Calculation of temperature from the kinetic energy.

virial.F : Routine for the calculation of the intermolecular virial (total forces on molecules, acting on centers of mass, are used).

virial12.F : Calculation of pressure from intermolecular virial.

widom.F : Test particle insertions on configurations read from tape file.

wtime.F : Random number generation, timing.

Brief list of post-processing routines for dynamical properties (subdirectory Post)

driver.F : coordination of calculations.

corrfun.F : Velocity autocorrelation function and its Fourier transform.

cosfft.F : Cosine fast Fourier transform routine.

filon.F : Cosine Fourier transform by Filon's method.

period.F : Imposition of periodic boundary conditions.

rotate.F : Computes correlation functions for rotational motion. (Don't worry about it).

schoen.F : Self-diffusivity calculation from mean squared displacement.

Brief list of post-processing routines for the calculation of liquid structure (subdirectory GofR)

driver.F : Main program coordinating the calculations. (You will have to modify this!)

grov.F : Routine for constructing the histogram and calculating radially averaged pair distribution functions for molecular centers. (You will have to write this!)

grov2.F : Routine for normalizing the histograms and writing pair distribution function to an output file. (You will have to write this!).

setup.F : Reads in control file and performs initializations.

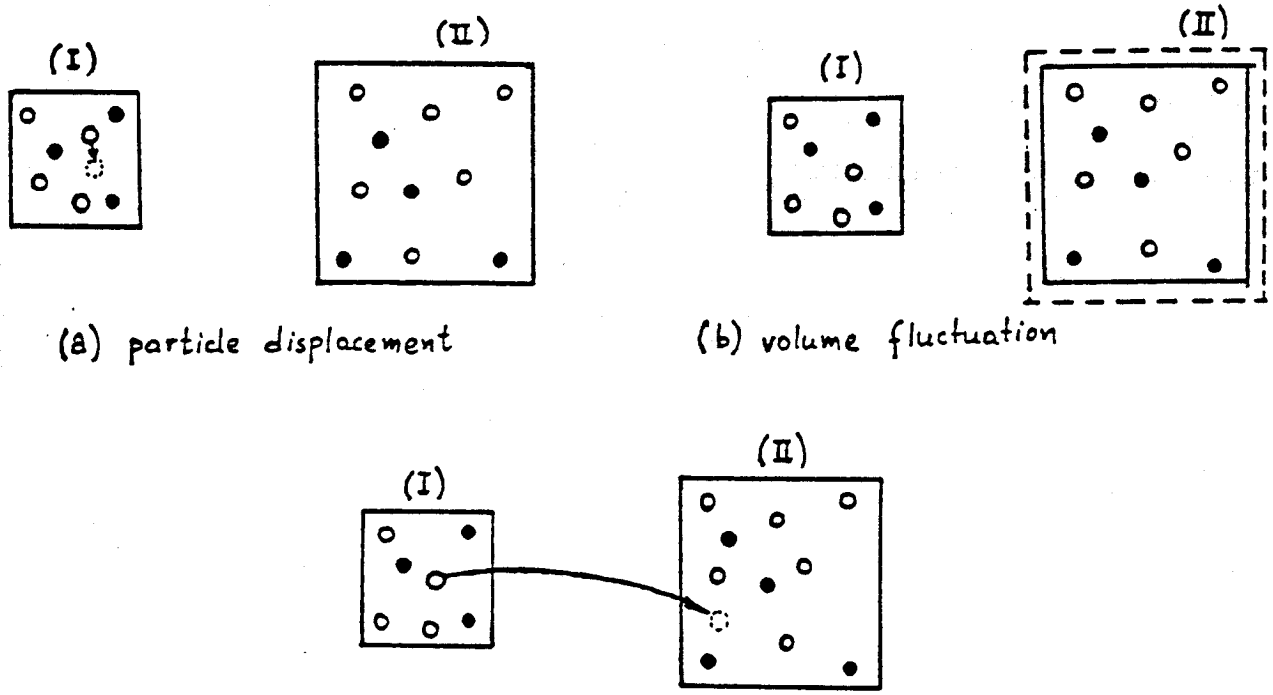
wtime.F : Random number generation, timing.

Problem II: Calculation of two-phase equilibrium in a binary system by Gibbs ensemble Monte Carlo simulation

A.Z. Panagiotopoulos introduced a new Monte Carlo technique that is particularly convenient for phase equilibrium calculations. He termed this technique "Gibbs ensemble Monte Carlo." A good description of the technique and its theoretical basis can be found in Panagiotopoulos, A.Z., Quirke, N., Stapleton, M., Tildesley, D.J. (1988). *Mol. Phys.* **63**, 527-545.

The constant pressure version of the Gibbs ensemble Monte Carlo method can be envisioned as a hybrid between the isothermal-isobaric and Grand canonical techniques. Consider the equilibrium between two phases (I and II; *e.g.*, gas and liquid) of a binary system of two components, 1 and 2. The number of degrees of freedom of such a system is $C+2-P=2$; the intensive state of the two-phase system is specified completely by fixing temperature T and pressure P. For example, the densities ρ^I, ρ^{II} and mole fractions x_1^I, x_1^{II} are fixed once we specify T and P. The constant pressure Gibbs ensemble MC

simulates the two coexisting phases as bulk phases, i.e., without the explicit consideration of an interface. Each phase is represented by a box with periodic boundary conditions. Three types of moves are implemented (see figure):



(a) *Particle Displacement*

Attempts to displace a molecule take place independently in each of the two boxes as in canonical ensemble MC. In this move, one of the boxes is chosen at random with a pre-specified probability. A molecule in the chosen box is chosen at random and subjected to a random displacement, whose components are uniformly distributed within an interval $(-\delta r_{max}, \delta r_{max})$. Such a trial move is accepted with probability

$$\mathcal{P}^J_{move} = \min\left(\frac{\exp\left[-\beta \mathcal{V}^J_{new}\right]}{\exp\left[-\beta \mathcal{V}^J_{old}\right]}, 1\right) = \min\left(1, \exp\left(-\beta \Delta \mathcal{V}^J\right)\right)$$

where $J = I$ or II the chosen box, exactly as in canonical MC.

(b) *Volume fluctuation*

Volume fluctuation moves are also performed independently in the two boxes. A box is selected at random with prespecified probability, and its edge length is expanded or contracted by a length picked at random from a uniform distribution within the interval $(-\delta L_{max}, \delta L_{max})$. All molecular centers in the deformed box are affinely displaced during this move, as in NPT Monte Carlo. The acceptance probability dictated by the Metropolis criteria is

$$\mathcal{P}_{vol}^{J} = \min\left\{1, \exp\left[-\beta\left(\Delta\mathcal{V}^{J} + P\Delta V^{J} - N^{J}k_{B}T\ln\frac{V_{new}^{J}}{V_{old}^{J}}\right)\right]\right\}$$

where J the chosen box and N^J the total number of molecules in it.

(c) *Particle interchange*

This move involves both boxes, I and II. Whereas move (a) can be viewed as imposing the condition of constant temperature, T, and move (it b) as imposing the condition of constant pressure, P, on the system, the objective of the particle interchange move is to impose the condition of equality of chemical potentials:

$$\mu_1^I = \mu_1^{II}\ ,\ \ \mu_2^I = \mu_2^{II}$$

A particle interchange move consists of the annihilation of a molecule in one phase ("origin" box), with simultaneous creation of a molecule of the same species in the other phase ("destination" box). It is implemented as follows:

- Choose either box I or II for the particle creation, with *equal* probability.

- Select with a fixed (but otherwise arbitrary) probability which species will be interchanged.
- Select randomly one particle representative of the chosen species to be destroyed in the origin box. Calculate the energy change associated with the particle destruction in the origin box, $\Delta\mathcal{V}^{origin}$.
- Select randomly a position for the particle to be inserted in the destination box. Calculate the energy change associated with the particle insertion at this position, $\Delta\mathcal{V}^{dest}$.

The acceptance probability arrived at by consideration of the probability density in configuration space of the two-phase system is

$$\mathcal{P}_{xch} = \min\left\{1, \exp\left[-\beta\left(\Delta\mathcal{V}^{dest} + \Delta\mathcal{V}^{orig} + k_B T \ln\frac{V^{orig}(N^{dest}+1)}{V^{dest}N^{orig}}\right)\right]\right\}$$

In this problem, you are asked to predict the gas-liquid coexistence behavior of a simple binary mixture of spherical Lennard-Jones molecules through constant pressure Gibbs ensemble Monte Carlo simulations. Assume the following potential parameters for the mixture; these parameters have been proposed in the context of a crude model for acetone (1)/ carbon dioxide (2):

	ϵ_{ij}/k_B (K)			σ_{ij}(Å)	
	1	2		1	2
1	377	291	1	4.95	4.38
2	291	225	2	4.38	3.80

By conducting constant pressure Gibbs ensemble MC simulations of this mixture at temperature of 350K and in the pressure range 30 to 90bar, you are asked to calculate and plot:

(a) The *Pxy* diagram for this model mixture at 350K (*i.e.*, a diagram with abscissa the saturated vapor or liquid composition and ordinate the pressure).

(b) The molar density of the vapor and liquid phases as a function of pressure at 350K.

FORTRAN codes for the Gibbs ensemble simulation are provided within the subdirectory *Gibbs* of the *student* directory. You are given subroutines which perform all the elementary operations involved in the simulation. [*You will have to write the driver program* (let us call it GIBBS.F) to coordinate the given routines and extract the required information from your simulations.] In general, quantities in the routines are expressed in dimensionless form. Energies are reduced by ϵ_{11}, lengths by σ_{11}, temperatures by ϵ_{11}/k_B, pressures by $\epsilon_{11}/\sigma_{11}^3$, and molecular densities by $1/\sigma_{11}^3$. The routines in your directory have been written by DNT with the help of Larry R. Dodd, using some codes provided by Allen and Tildesley as a starting point. The routines are well documented and simple in structure. A makefile for compilation (assuming you have written the driver program GIBBS.F) is provided.

Brief list of the Gibbs ensemble Monte Carlo routines

ADD.F : Updates the configuration and energy of a box upon acceptance of a move that resulted in the addition of a molecule in that box.

BOXCONFIG.F : Creates an initial guess configuration for the contents of a box, in the form of a defective *fcc* lattice, by calling the routine FCC.F.

ENERGY.F : Computes the potential energy felt by a molecule as a result of its interactions with the rest of a box, including long-range (tail) contributions. Must be used to obtain the energy changes associated with molecule displacement and interchange moves.

FCC.F Sets up an FCC lattice of spherical Lennard-Jones molecules.

MOVE.F : Performs a molecule displacement move, accepts/rejects it, and updates the configuration accordingly.

PRINT.F : Computes current ensemble averages and variances from property accumulators and prints detailed and concise information on the progress of the run.

PRINTSTAT.F : Prints out statistics on the rate of acceptance of attempted moves at the end of a run.

RANDOM.F : Pseudorandom number generator.

SUMUP.F : Computes the total potential energy and virial of a phase, given the configuration.

UPDATE.F : Updates accumulators of averages and variances since a specified step up to the current simulation step.

READCN.F : Reads starting configuration from a binary file written by routine WRITECN.F. (Should be useful in continuing runs from a given set of conditions to another).

VOLFLUC.F : Performs volume fluctuation move, assesses its energetic consequences, accepts/rejects it, and updates the configuration accordingly.

REMOVE.F : Updates the configuation and energy of a box upon acceptance of a move that resulted in the removal of a molecule from the box.

WRITECN.F : Saves the current configuration in a binary (crash) file for future use.

START.F : Coordinates the creation of initial configurations for the two boxes by invoking the routine BOXCONFIG.

XCHNG.F : Performs a molecule interchange movem accepts/rejects it, and updates the configuration accordingly.

Happy Computing!