

Code examples

The accompanying code is provided to show examples of some of the techniques and algorithms described in our book “Nonequilibrium Molecular Dynamics: Theory, Algorithms and Applications” by Billy D. Todd and Peter J. Daivis (Cambridge University Press 2017).

The nemd program and the accompanying input files can be used to generate some of the data that we have provided in the book for validation purposes. While we have taken some care to eliminate obvious bugs, we provide no assurance that the code will function correctly for all combinations of input parameters. Some optimisation has been done, but it is not highly optimised, nor is it parallelised. Another version of the code that uses the Fortran 2008 coarray functionality for ensemble averaging implements parallelisation using a very simple strategy will be published elsewhere.

The main objective here has been to maintain flexibility and ease of use for pedagogical and research purposes. We encourage users to experiment with the code, as we have done over many years, and enjoy using it to learn about nonequilibrium phenomena. We ask only that if it is used for published research, our book is cited and the source of the code acknowledged appropriately as an accompaniment to the book. Commercial use without the authors’ permission is prohibited.

The nemd program is written in modern Fortran (extensively using features of Fortran 90, and revisions of the standard up to Fortran 2015) and is designed to perform equilibrium and nonequilibrium molecular dynamics computations for a few commonly used molecular models. The focus is on computation of transport coefficients such as the viscosity, diffusion coefficients and thermal conductivity. It is limited in capability and is not intended to be a comprehensive package, but it could be used as a teaching and learning tool for those who are interested in homogeneous nonequilibrium molecular dynamics simulations. The building blocks are provided in the form of a set of quite flexible modules that can be adapted as required.

Contents of the archive

bin - executable binary files

build - intermediate files produced by the compiler

doc - miscellaneous documentation including a copy of this document

makefile - makefile for compiling the executable

run_scripts - simple scripts containing commands to run the executable

source - source code

nemd_test_input - a set of example input files that can be used for testing

Program units in the source directory

nemd.f90	main program
atoms.f90	module defining data structure and related procedures for positions, momenta and force on each atom
barostat.f90	module defining data structure for barostat used in constant pressure simulations
bond_bend.f90	module defining data structures and related procedures for bond angle bending potential
bond_stretch.f90	module defining data structures and related procedures for bond stretching potentials
constraints.f90	module defining data structures and related procedures for bond length and bond angle constraints using Gaussian constraint algorithm
dihedral_force.f90	module defining data structures and related procedures for simple dihedral potential
ensemble_av_nocoarray.f90	module defining data structures and related procedures for ensemble averaging over parallel images using the Fortran 2008 coarray functionality. This has been disabled in the current version of the code.
ewald.f90	module defining data structures and related procedures for electrostatic interactions using the Ewald summation method
gear5.f90	module defining data structures and related procedures for integration of first order equations of motion using the Gear predictor-corrector algorithm
integrator.f90	module defining data structures and related procedures for integration of the equations of motion. Uses specific integrators are required.
lattice.f90	module defining data structures and related procedures for constructing initial configuration of atoms or molecules in an fcc lattice

mdstep.f90	module containing procedures to carry out a single md step
molecules.f90	module defining data structures and related procedures for molecular properties and characteristics. Calculates molecular centres of mass, total momentum, force and energy per molecule etc.
neighbour_list.f90	module defining data structures and related procedures for constructing neighbour list for pair interactions
pair_force.f90	module defining data structures and related procedures for calculations of short ranged intermolecular pair forces and energies for a few simple potentials, including the Lennard-Jones potential.
pduil.f90	module containing file initialisation procedures
properties.f90	module defining data structures and related procedures for managing property computation, accumulation and storage
run_control.f90	module defining data structures and related procedures for run control variables, including total steps for the run, input and output file management.
simulation_box.f90	module defining data structures and related procedures for managing the simulation box including periodic boundary conditions
slld.f90	module defining data structures and related procedures for nonequilibrium simulations of shear flow, planar elongation and bulk isotropic deformation (compression or expansion) with the slld equations of motion
species.f90	module defining data structures and related procedures for managing atomic species characteristics such as mass, charge and labels
system.f90	module defining data structures and related procedures for managing the thermal system, including temperature and density initialisation
tcf.f90	module defining data structures and related procedures for computing and managing time correlation functions
thermostat.f90	module defining data structures and related procedures for computing and controlling temperature

v_sub_library.f90 module defining customised procedures used for
 mathematical computations including vectorised solution
 of linear equations

Test input files and test output

For an overview of the capabilities of the code and a staged introduction to its operation, it is recommend that the input files are tested and run in the following order:

WCA

LJ_2.5

Ar_Kr_mixture

Chlorine

Carbon_disulphide

Butane_RB_model

Colloid

SPCE

SPCE_NaCl

10_site_fjc

fjc_binary_10_50_site

The enclosed README.txt files in each directory contain a description of the simulated system and other important information.