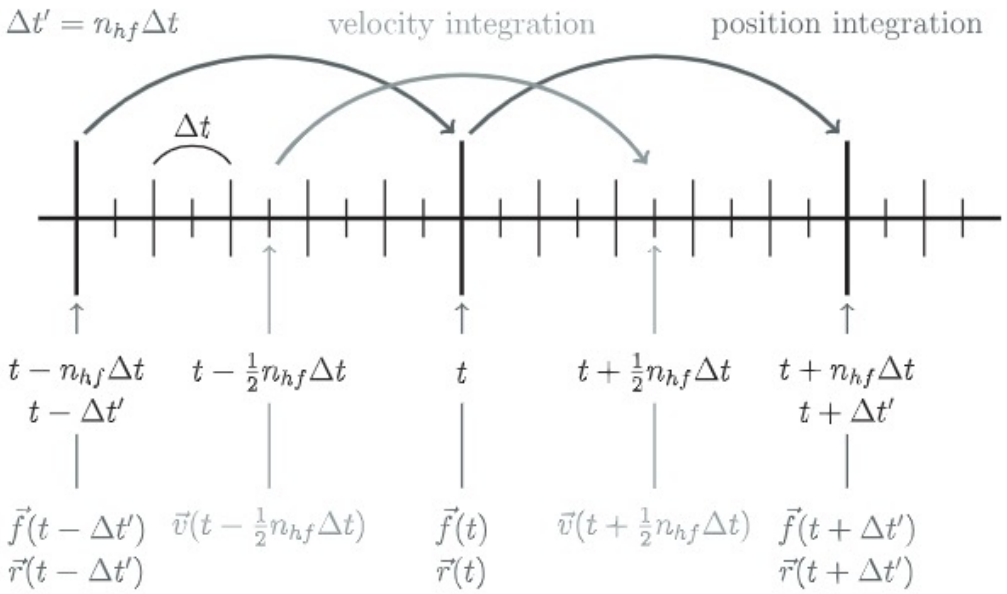
Integration Algorithm



2024 Winter Seminar

January 12th, 2024

Sangmin Lee

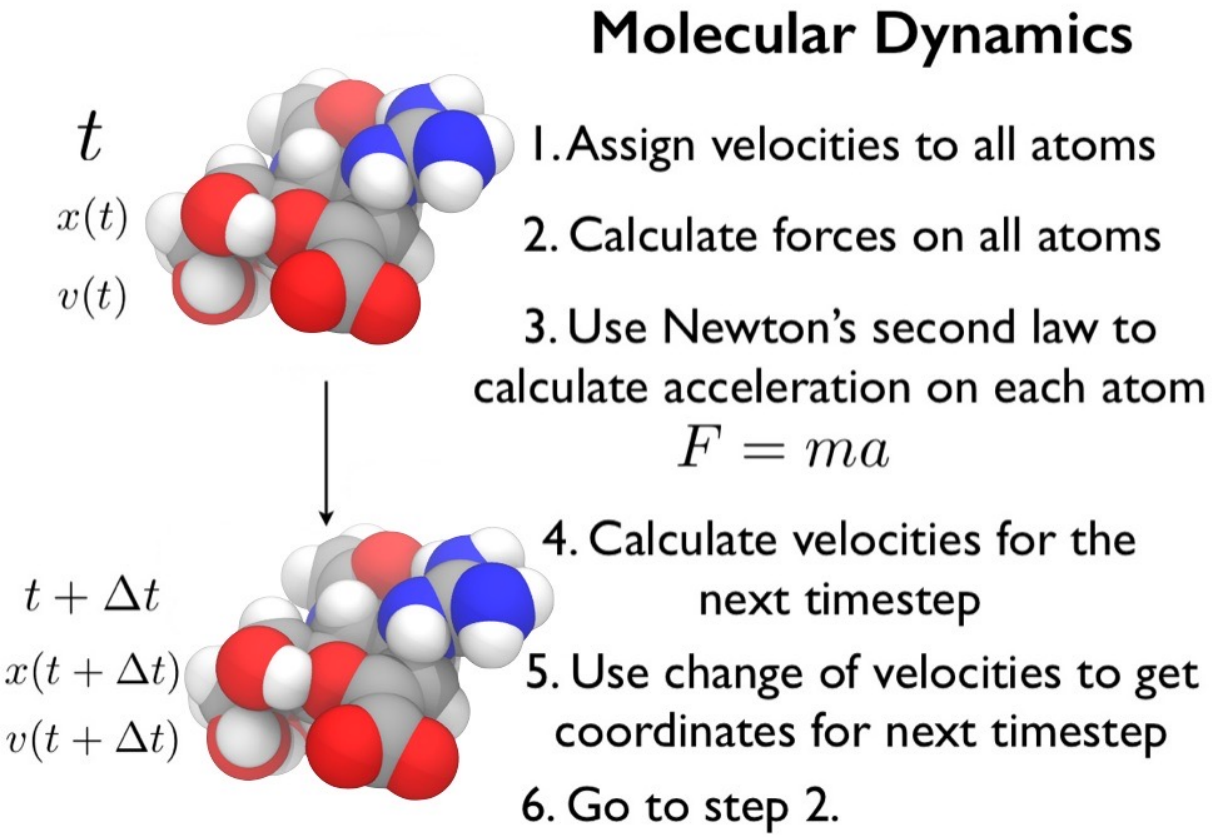
Content

Introduction

Theoretical background

Journey to increase the time step

Yes we solve F=ma in classical MD simulation



But there are much more things that we should understand here!

The concept of Integrator

In OpenMM,

`LangevinIntegrator`	This is an Integrator which simulates a System using Langevin dynamics.
`LangevinMiddleIntegrator`	This is an Integrator which simulates a System using Langevin dynamics, with the LFMiddle discretization (J.
`MTSIntegrator`	MTSIntegrator implements the rRESPA multiple time step integration algorithm.
`MTSLangevinIntegrator`	MTSLangevinIntegrator implements the BAOAB-RESPA multiple time step algorithm for constant temperature dynamics.
`NoseHooverIntegrator`	This is an Integrator which simulates a System using one or more Nose Hoover chain thermostats, using the "middle" leapfrog propagation algorithm described in J.
`RPMDIntegrator`	This is an Integrator which simulates a System using ring polymer molecular dynamics (RPMD).
`VariableLangevinIntegrator`	This is an error controlled, variable time step Integrator that simulates a System using Langevin dynamics.
`VariableVerletIntegrator`	This is an error controlled, variable time step Integrator that simulates a System using the leap-frog Verlet algorithm.
`VerletIntegrator`	This is an Integrator which simulates a System using the leap-frog Verlet algorithm.

In OpenMMtools (https://openmmtools.readthedocs.io/en/stable/integrators.html),

`LangevinIntegrator`	Integrates Langevin dynamics with a prescribed operator splitting.
`VVVRIntegrator`	Create a velocity Verlet with velocity randomization (VVVR) integrator.
`BAOABIntegrator`	Create a BAOAB integrator.
`GeodesicBAOABIntegrator`	Create a geodesic-BAOAB integrator.
`GHMCIntegrator`	Create a generalized hybrid Monte Carlo (GHMC) integrator.

`NonequilibriumLangevinIntegrator`	Nonequilibrium integrator mix-in.
`AlchemicalNonequilibriumLangevinIntegrator`	Allows nonequilibrium switching based on force parameters specified in alchemical_functions.
`PeriodicNonequilibriumIntegrator`	Periodic nonequilibrium integrator where master alchemical parameter `lambda` is driven through a periodic protocol:
`ExternalPerturbationLangevinIntegrator`	Create a LangevinSplittingIntegrator that accounts for external perturbations and tracks protocol work.

`MTSIntegrator`	MTSIntegrator implements the rRESPA multiple time step integration algorithm.
`DummyIntegrator`	Construct a dummy integrator that does nothing except update call the force updates.
`GradientDescentMinimizationIntegrator`	Simple gradient descent minimizer implemented as an integrator.
`VelocityVerletIntegrator`	Verlocity Verlet integrator.
`AndersenVelocityVerletIntegrator`	Velocity Verlet integrator with Andersen thermostat using per-particle collisions (rather than massive collisions).
`NoseHooverChainVelocityVerletIntegrator`	Nosé-Hoover chain thermostat, using the reversible multi time step velocity Verlet algorithm
`MetropolisMonteCarloIntegrator`	Metropolis Monte Carlo with Gaussian displacement trials.
`HMCIntegrator`	Hybrid Monte Carlo (HMC) integrator.

Basics of MD simulation

In classical mechanics, the equation of motion is integrated to generate the trajectory.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{r}}_i}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{r}_i} = 0. \qquad \dot{q}_\alpha = \frac{\partial \mathcal{H}}{\partial p_\alpha}, \;\; \dot{p}_\alpha = -\frac{\partial \mathcal{H}}{\partial q_\alpha}.$$

Energy is conserved in classical mechanics!

By solving F=ma, we sample the **microcanonical** ensemble (constant E) for ensemble averages.

$$\langle a \rangle = \frac{\int \mathrm{dx}\, a(\mathrm{x})\delta(\mathcal{H}(\mathrm{x}) - E)}{\int \mathrm{dx}\, \delta(\mathcal{H}(\mathrm{x}) - E)} = \lim_{\mathcal{T}\to\infty} \frac{1}{\mathcal{T}} \int_0^{\mathcal{T}} \mathrm{d}t\, a(\mathrm{x}_t) \equiv \bar{a}.$$

We need to sample the x_t in microcanonical ensemble, which we call it **trajectory**.(.dcd)

In here, we introduce the time discretization parameter dt, known as the **time step.**

Starting with the initial cond x_0, x_dt, x_2dt, x_3dt are generated by applying the integrator iteratively.

$$A = \langle a \rangle = \frac{1}{M} \sum_{n=1}^{M} a(\mathrm{x}_{n\Delta t}) \equiv \bar{a}.$$

The very first integration algorithm : Verlet Algorithm (1967)

PHYSICAL REVIEW VOLUME 159, NUMBER 1 5 JULY 1967

Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules*

LOUP VERLET†
Belfer Graduate School of Science, Yeshiva University, New York, New York
(Received 30 January 1967)

The equation of motion of a system of 864 particles interacting through a Lennard-Jones potential has been integrated for various values of the temperature and density, relative, generally, to a fluid state. The equilibrium properties have been calculated and are shown to agree very well with the corresponding properties of argon. It is concluded that, to a good approximation, the equilibrium state of argon can be described through a two-body potential.

Let's use a Taylor series to the position of a particle at time t+dt and t-dt.

$$\mathbf{r}_i(t+\Delta t) \approx \mathbf{r}_i(t) + \Delta t \dot{\mathbf{r}}_i(t) + \frac{1}{2}\Delta t^2 \ddot{\mathbf{r}}_i(t) \approx \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t).$$

$$\mathbf{r}_i(t-\Delta t) = \mathbf{r}_i(t) - \Delta t \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t).$$

Adding both two equations, one obtains

$$\mathbf{r}_i(t+\Delta t) + \mathbf{r}_i(t-\Delta t) = 2\mathbf{r}_i(t) + \frac{\Delta t^2}{m_i}\mathbf{F}_i(t),$$

After rearrangement,

$$\mathbf{r}_i(t+\Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t) + \frac{\Delta t^2}{m_i}\mathbf{F}_i(t).$$

Phys. Rev. ***1967***, 159, 98.

The list of constraint algorithms

Let's use a Taylor series to the position of a particle at time t+dt and t-dt.

In OpenMM,

SETTLE for water molecules only.

SHAKE for isolated clusters of one heavy atom with up to three hydrogens bonded to it.

CCMA for anything not handled by one of the above algorithms.

(LINCS is not implemented in OpenMM)

There are two anomalies, probably both related to the LINCS constraint algorithm (27) in GROMACS. Although we could implement all of Martini 2 and Martini 3 in OpenMM, the commonly used Martini 2 topology for cholesterol is unstable in OpenMM. We traced this back to the definition of constraints in the virtual sites used in cholesterol. These constraints are deemed solved with LINCS, although the solution is inaccurate, and the OpenMM algorithm cannot satisfy the constraints. We developed a modified cholesterol topology for Martini 2 (Fig. S1 and Supporting material: "Development of a modified cholesterol topology for OpenMM") that uses a different set of constraints and a total of four virtual sites, which improves the geometry and sat-

Biophys. J. ***2023***, 122, 14, 2864-2870.

Convergence issues on constraint algorithm

Let's use a Taylor series to the position of a particle at time t+dt and t-dt.

$$\mathbf{r}_i(t+\Delta t) \approx \mathbf{r}_i(t) + \Delta t\dot{\mathbf{r}}_i(t) + \frac{1}{2}\Delta t^2\ddot{\mathbf{r}}_i(t) \approx \mathbf{r}_i(t) + \Delta t\mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t).$$

$$\mathbf{r}_i(t-\Delta t) = \mathbf{r}_i(t) - \Delta t\mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t).$$

Adding both two equations, one obtains

$$\mathbf{r}_i(t+\Delta t) + \mathbf{r}_i(t-\Delta t) = 2\mathbf{r}_i(t) + \frac{\Delta t^2}{m_i}\mathbf{F}_i(t),$$

After rearrangement,

$$\mathbf{r}_i(t+\Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t) + \frac{\Delta t^2}{m_i}\mathbf{F}_i(t).$$

Phys. Rev. ***1967***, 159, 98.

Shot time step is sometimes inevitable

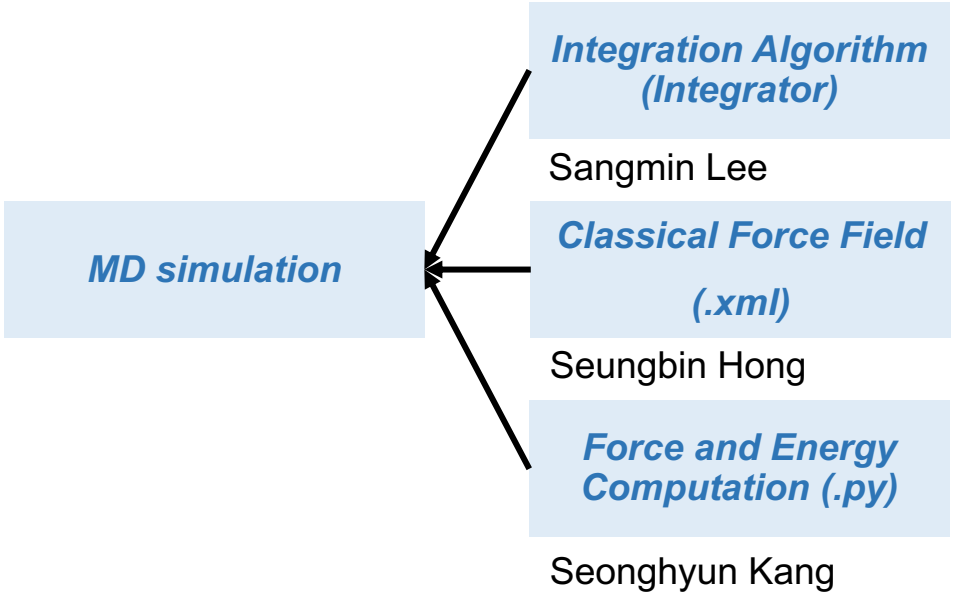
$$\mathbf{r}_i(t+\Delta t) \approx \mathbf{r}_i(t) + \Delta t \dot{\mathbf{r}}_i(t) + \frac{1}{2}\Delta t^2 \ddot{\mathbf{r}}_i(t) \approx \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t).$$

$$\mathbf{r}_i(t-\Delta t) = \mathbf{r}_i(t) - \Delta t \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t).$$

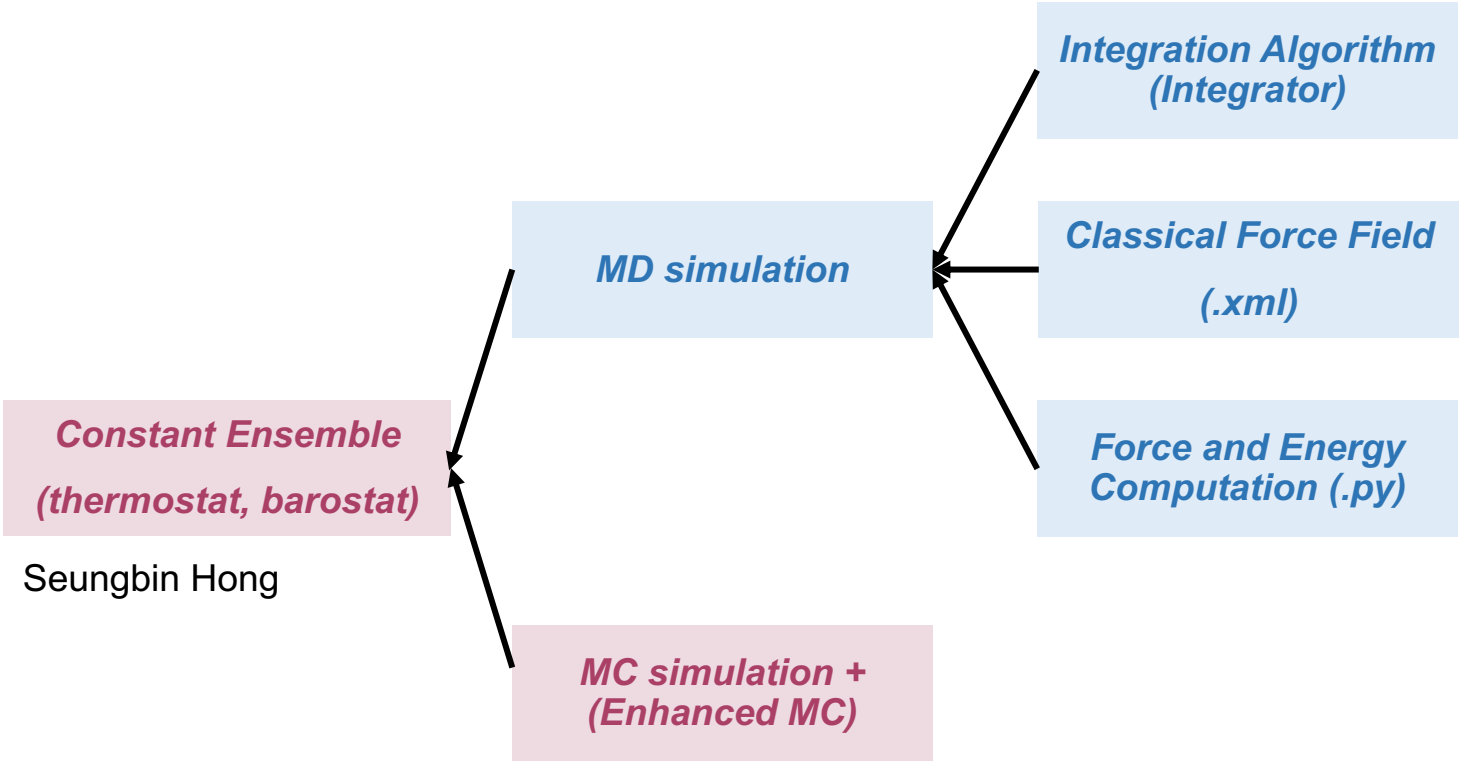
$$\mathbf{r}_i(t+\Delta t) + \mathbf{r}_i(t-\Delta t) = 2\mathbf{r}_i(t) + \frac{\Delta t^2}{m_i}\mathbf{F}_i(t),$$

$$\mathbf{r}_i(t+\Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t) + \frac{\Delta t^2}{m_i}\mathbf{F}_i(t).$$

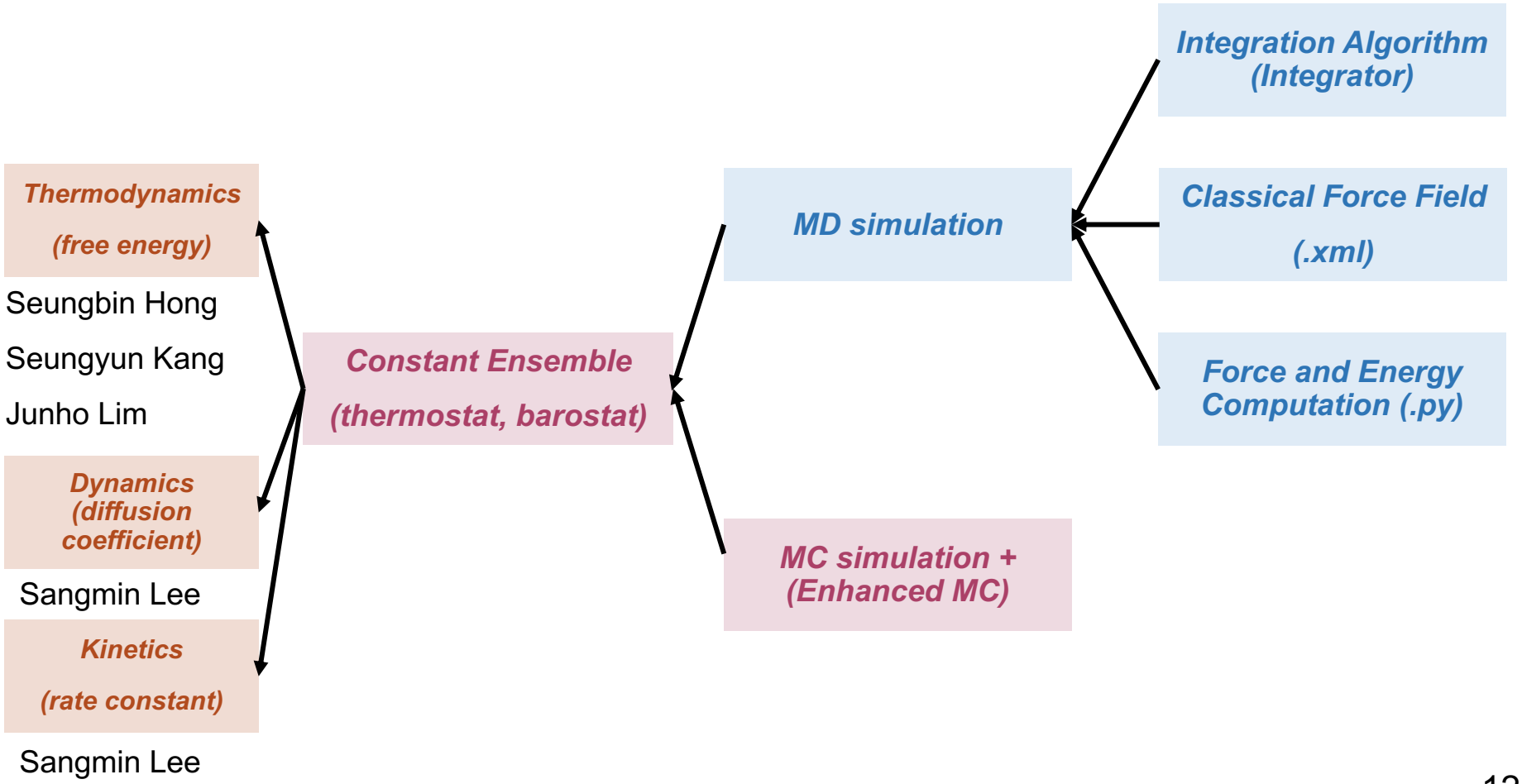
Phys. Rev. ***1967***, 159, 98.


MD simulation
Integration Algorithm (Integrator)
Sangmin Lee
Classical Force Field (.xml)
Seungbin Hong
Force and Energy Computation (.py)
Seonghyun Kang

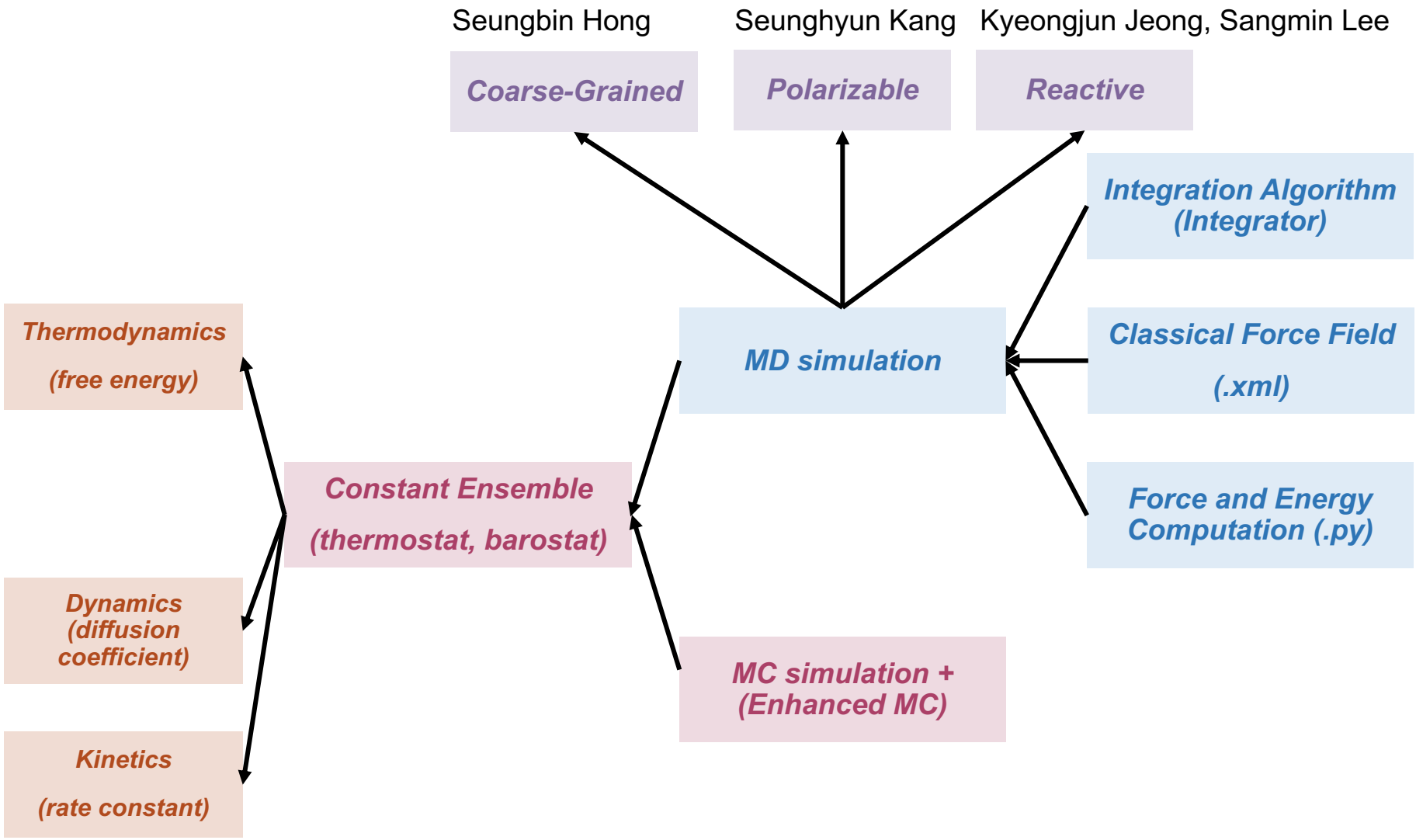
Week2 – How to sample ensembles?



Week3 – What to compute?



Week4 – How to extend the scope?




일어나
세미나
준비
해야지