CS 6014: Computation as a Research Tool

Atomistic Simulation Track

Instructor: Leonid Zhigilei, Department of Materials Science & Engineering

Lecture 1 (04/02): Brief introduction, basic concepts of molecular dynamics (MD), Verlet and Velocity Verlet algorithm for equations of motion, homework #1.

Lecture 2 (04/04): More detailed discussion of MD, assumptions and limitations of classical MD.

Lecture 3 (04/06): Examples of applications of MD, discussion of projects suggested by students.

Lecture 4 (04/09): Overview of Monte Carlo Methods, basic ideas of Metropolis Monte Carlo method.

Lecture 5 (04/11): Comparative analysis of the capabilities of kinetic Monte Carlo, Metropolis Monte Carlo, and molecular dynamics methods, homework #1 is due

Lecture 6 (04/13): Discussion of the results of homework #1, discussion of projects suggested by students.

Instructor: Matthew Neurock, Department of Chemical Engineering

Lectures 7 - 12: topics to be announced, Homework #2

Outline of Lecture 1

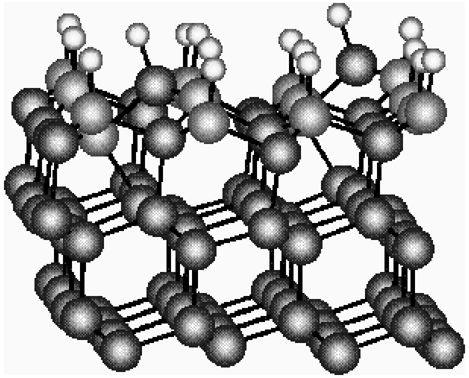
- Introduction
  - **Computational methods in chemistry, materials science & mechanical engineering**
- Quick introduction to molecular dynamics
  - **Molecular Dynamics – direct simulation of atomic motions**
  - **Integration of the equations of motion – Verlet and Velocity Verlet algorithm**
  - **Interatomic potentials – Lennard Jones potential**
- Discussion of Homework #1

Textbooks on Molecular Dynamics:

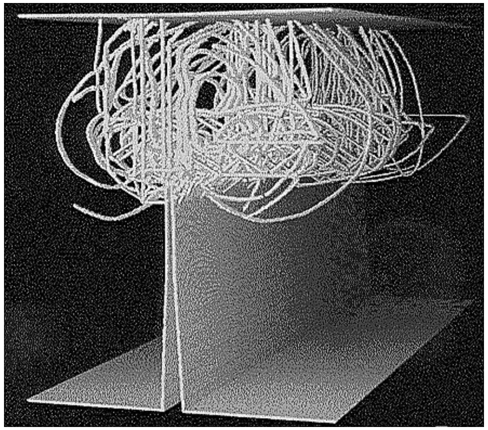
1. M. P. Allen, D. J. Tildesley, Computer simulation of liquids (Clarendon Press: Oxford, 1990).
2. D. Frenkel, B. Smit, Understanding molecular simulation from algorithms to applications (Academic Press: San Diego, 1996).

Length- and time-scales in materials modeling

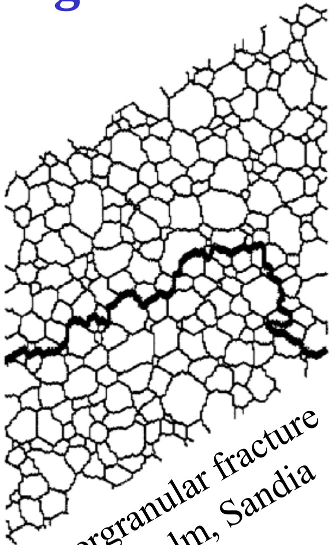
Computational Chemistry



structure of diamond surface

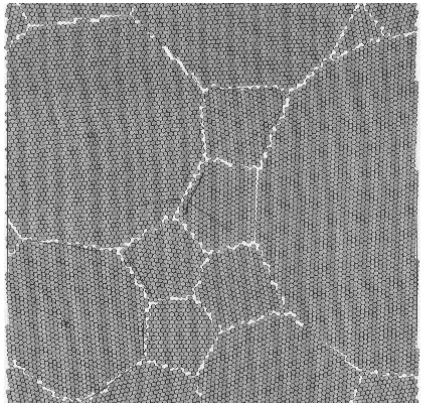


crack propagation, F. F. Abraham, IBM

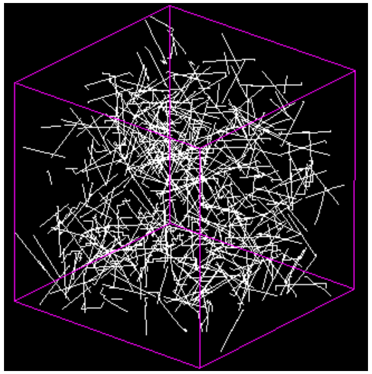


intergranular fracture
E. Holm, Sandia

Computational Mechanics



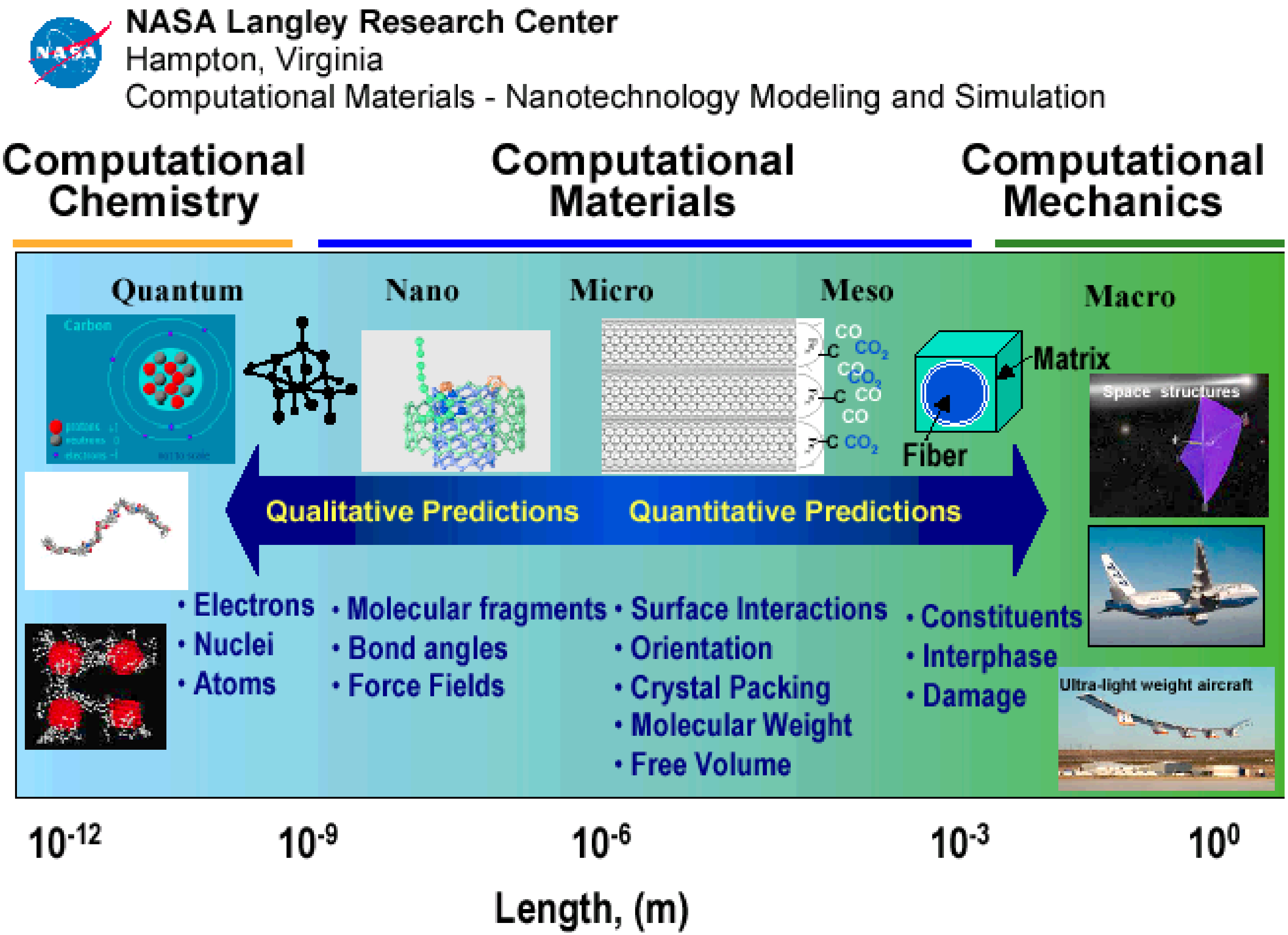
Nanocrystalline material, M. Li, JHU



plastic deformation of a crystal
http://zig.onera.fr/DisGallery/index.html

10^{3}	10^{6}	10^{9}	Length Scale, number of atoms	10^{21}
10^{-9}	10^{-8}	10^{-7}	Length Scale, meters	10^{-3}
10^{-12}	10^{-9}	10^{-7}	Time Scale, seconds	1

Length and time scales in materials modeling *(by Greg Odegard, NASA)*



NASA Langley Research Center
Hampton, Virginia
Computational Materials - Nanotechnology Modeling and Simulation

Computational Chemistry

Computational Materials

Computational Mechanics

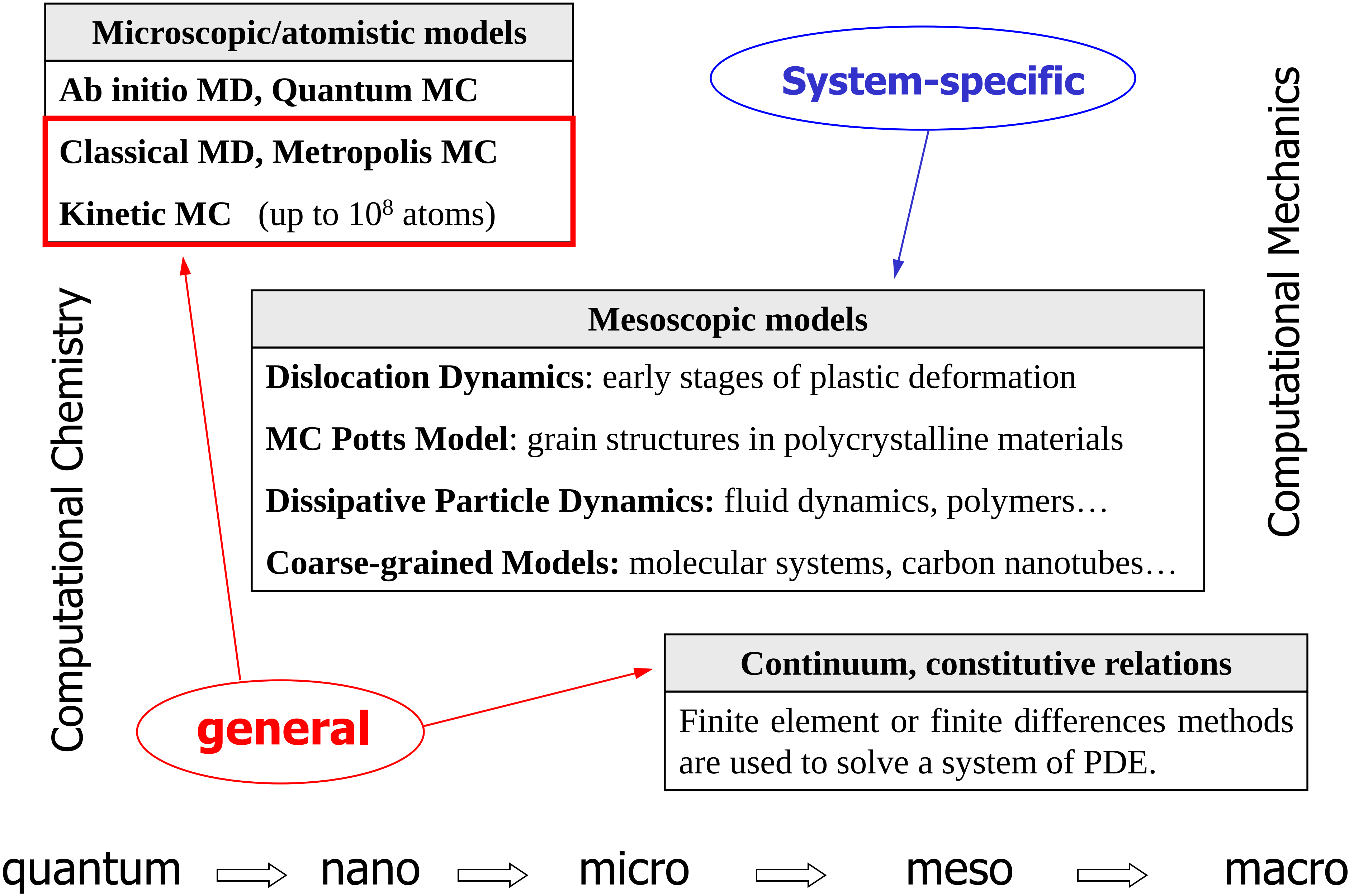
- Electrons
- Nuclei
- Atoms

- Molecular fragments
- Bond angles
- Force Fields

- Surface Interactions
- Orientation
- Crystal Packing
- Molecular Weight
- Free Volume

- Constituents
- Interphase
- Damage

Computational methods in materials science



Microscopic/atomistic models

Ab initio MD, Quantum MC

Classical MD, Metropolis MC

Kinetic MC (up to 10^8 atoms)

System-specific

Mesoscopic models

Dislocation Dynamics: early stages of plastic deformation

MC Potts Model: grain structures in polycrystalline materials

Dissipative Particle Dynamics: fluid dynamics, polymers…

Coarse-grained Models: molecular systems, carbon nanotubes…

Continuum, constitutive relations

Finite element or finite differences methods are used to solve a system of PDE.

general

Computational Chemistry

Computational Mechanics

quantum ⇨ nano ⇨ micro ⇨ meso ⇨ macro

Molecular dynamics method

Molecular dynamics (MD) – computational method for simulation of the time evolution of a system of interacting particles (atoms, molecules, granules, etc.).

The basic idea is simple:

First, for a system of interest, one has to specify:

- **initial conditions** (initial positions & velocities of all particles in the system)
- **interaction potential** for deriving the forces among all the particles

Second, the evolution of the system in time is followed by **solving a set of classical equations of motion** for all particles in the system.

$$m_i \vec{a}_i = \vec{F}_i \quad \text{or} \quad m_i \frac{d^2 \vec{r}_i}{dt^2} = \vec{F}_i \qquad \text{for the } i^{th} \text{ particle}$$

$$\vec{F}_i = -\vec{\nabla}_i U(\vec{r}_1, \vec{r}_2, \vec{r}_3, ..., \vec{r}_{N_{at}})$$

Output: positions and velocities of all the atoms as a function of time, $\vec{r}_i(t), \vec{v}_i(t)$

Advantages and limitations of molecular dynamics

Advantages of MD:

- the only input in the model – description of interatomic/intermolecular interaction
- no assumptions are made about the processes/mechanism to be investigated
- provides a detailed molecular/atomic-level information

Can discover new physics/mechanisms!

Limitations of MD: - will discuss in Lecture 2

- classical description of interatomic interaction
- classical description of atomic motion
- small length scales (up to 100s of nm), short times (up to 100s of ns)
- it is often difficult to provide a quantitative description of real materials (even with “first-principles” methods)

Molecular Dynamics → Ordinary Differential Equations (ODE)

Consider system of N particles. Trajectory $\mathbf{r}_i(t)$ of a particle i of mass of m_i, under the action of force $\mathbf{F}_i$ can be described by the Newton's equation of motion - the second order differential equations:

$$m_i \frac{d^2 \vec{r}_i(t)}{dt^2} = \vec{F}_i(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N, t)$$

in 3D space $\vec{r}_i = \{x_i, y_i, z_i\}$, $\vec{F}_i = \{F_i^x, F_i^y, F_i^z\}$, $i = 1,2,\ldots,N$

3N equations: $m_i \frac{d^2 x_i(t)}{dt^2} = F_i^x \qquad m_i \frac{d^2 y_i(t)}{dt^2} = F_i^y \qquad m_i \frac{d^2 z_i(t)}{dt^2} = F_i^z$

these 3N second order ODE can be rewritten as a system of 6N first order equations:

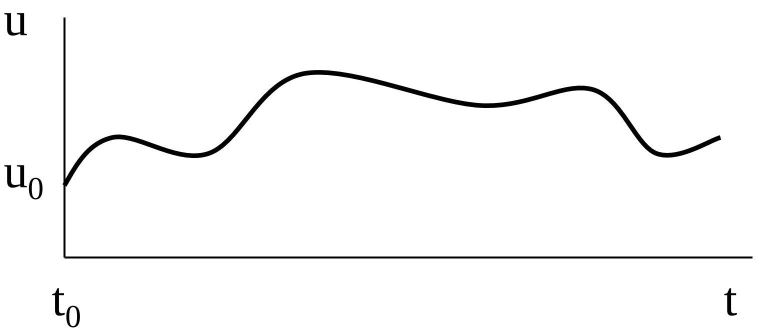
$$v_i^x = \frac{dx_i(t)}{dt}, \quad v_i^y = \frac{dy_i(t)}{dt}, \quad v_i^z = \frac{dz_i(t)}{dt}, \quad \frac{dv_i^x(t)}{dt} = \frac{F_i^x(t)}{m_i}, \quad \frac{dv_i^y(t)}{dt} = \frac{F_i^y(t)}{m_i}, \quad \frac{dv_i^z(t)}{dt} = \frac{F_i^z(t)}{m_i}$$

Therefore, in general, our task is to solve a system of equations du/dt = f(u,t) with initial conditions given by $u(t_0) = u_0$

This type of problem is called *initial value problem*

Finite difference methods for Particle Dynamics problem (discussed in ODE track)

Thus, we want to solve the following first order ODE equation $du/dt = f(u,t)$, $u(t_0) = u_0$



Finite difference method allows us, starting from the initial value u_0, to calculate values of $u(t_0+n\times h)$ one step at a time.

Finite difference technique:

1. Discretization of time: a grid of timesteps is introduced.
2. Replace the differential equation by the corresponding finite difference equation:

$$dt \rightarrow h$$
$$du \rightarrow \Delta u = u(t+h) - u(t)$$

$$\frac{du}{dt} = \lim_{h \to 0} \frac{u(t+h) - u(t)}{h} \approx \left.\frac{\Delta u}{h}\right|_{h \neq 0}$$

For $t_n = t_0 + n\times h$, $0 \leq n \leq (t_{last} - t_0)/h$ approximate solution is then $u_{n+1} = u_n + h*f(u_n, t_n)$

(Euler’s method)

Only slope of *u* at the beginning of the time interval is taken into account, the curvature is ignored – the Euler method is not accurate. Runge-Kutta methods account for the curvature, but at a cost of multiple evaluations of f(u,t) [forces in MD] at each time step – too computationally expensive and **never used in MD simulations**.

Special methods for particle dynamics

Second-order differential equations in which first-order derivatives do not appear are found so frequently in applied problems, that special methods have been devised for their solution.

$$m\frac{d^2 r(t)}{dt^2} = F(t)$$

The idea is to go directly from the second derivatives to the function itself without having to use the first order derivatives. Let's write two third-order Taylor expansions for r(t) at t+h and t-h and sum them together

$$r(t+h) = r(t) + h\frac{dr(t)}{dt} + \frac{h^2}{2}\frac{d^2 r(t)}{dt^2} + \frac{h^3}{6}\frac{d^3 r(t)}{dt^3} + O(h^4)$$

$$r(t-h) = r(t) - h\frac{dr(t)}{dt} + \frac{h^2}{2}\frac{d^2 r(t)}{dt^2} - \frac{h^3}{6}\frac{d^3 r(t)}{dt^3} + O(h^4)$$

$$\Longrightarrow \quad r(t+h) + r(t-h) = 2r(t) + h^2\frac{d^2 r(t)}{dt^2} + O(h^4)$$

$$\boxed{\mathbf{r(t+h) = -r(t-h) + 2r(t) + h^2F(t)/m + O(h^4)}}$$ - Verlet algorithm

Verlet algorithm is simple to implement, accurate and stable, time reversible

This algorithm is commonly used in MD simulations.

Minor problem: The velocity does not enter this algorithm explicitly. We need velocities to compute kinetic energy and to define the initial conditions (e.g. positions of atoms and T).

Velocity Verlet algorithm

Another implementation of the Verlet algorithm that explicitly includes velocities is so-called Velocity Verlet Algorithm.

$$\boxed{\begin{aligned} &\mathbf{r(t+h) = r(t) + h\,v(t) + h^2\,F(t)/2\,m + O(h^4)} \\ &\mathbf{v(t+h) = v(t) + h\,(F(t)/m + F(t+h)/m)/2 + O(h^3)} \end{aligned}}$$

- Self-starting from the positions and velocities at the initial time (does not require start-up calculation of r(t_0-h) as in the original Verlet).
- Mathematically identical to the original Verlet algorithm.
- Do not need to store the values of r(t) and v(t) at two different times.

Note that we have to calculate new forces F(t+h) after calculation of new positions r(t+h) but before calculation of new velocities → this can be a problem if forces depend on velocities.

You will need to implement the velocity Verlet algorithm in Homework #1.

Predictor-corrector algorithms that you discussed in the ODE track are also commonly used in MD simulations.

Interatomic interaction: Derivation forces for pair potential

In MD simulation we need forces that are acting on the atoms. The forces are given by the gradient of the potential energy surface (the force on atom *i* is a vector pointing in the direction of the steepest decent of the potential energy): $\vec{F}_i = -\vec{\nabla}_{r_i} U(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N)$

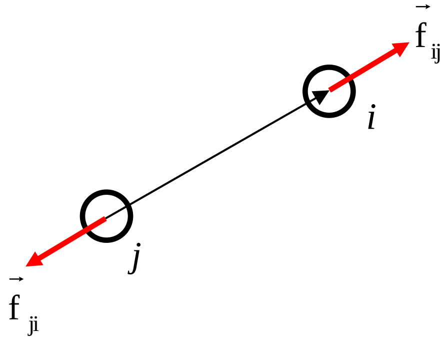
For a pair potential: $U(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N) = \sum_i \sum_{j>i} U(r_{ij})$ $\quad r_{ij} = \left|\vec{r}_i - \vec{r}_j\right| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$

The force on atom *i* is $\vec{F}_i = -\vec{\nabla}_{r_i} U(\vec{r}_1, \ldots, \vec{r}_i, \ldots, \vec{r}_N) = -\sum_{j \neq i} \vec{\nabla}_{r_i} U(r_{ij}) =$

$$= -\sum_{j \neq i} \left(\hat{x}\frac{\partial}{\partial x_i} + \hat{y}\frac{\partial}{\partial y_i} + \hat{z}\frac{\partial}{\partial z_i} \right) U(r_{ij}) = -\sum_{j \neq i} \left(\hat{x}\frac{\partial r_{ij}}{\partial x_i} + \hat{y}\frac{\partial r_{ij}}{\partial y_i} + \hat{z}\frac{\partial r_{ij}}{\partial z_i} \right) \frac{dU(r_{ij})}{dr_{ij}} =$$

$$= -\sum_{j \neq i} \left(\hat{x}\frac{x_{ij}}{r_{ij}} + \hat{y}\frac{y_{ij}}{r_{ij}} + \hat{z}\frac{z_{ij}}{r_{ij}} \right) \frac{dU(r_{ij})}{dr_{ij}} = -\sum_{j \neq i} \left(\frac{\vec{r}_i - \vec{r}_j}{r_{ij}} \right) \frac{dU(r_{ij})}{dr_{ij}} = \sum_{j \neq i} \vec{f}_{ij}$$

$$\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$$



$$U(r_{ij}) = U(r_{ji})$$

$$\vec{r}_{ij} = -\vec{r}_{ji}$$

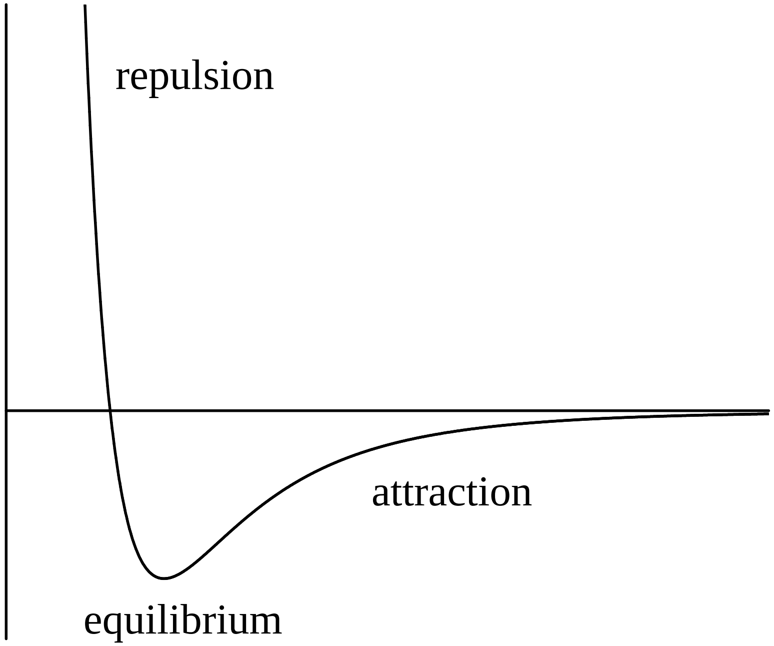
$$\vec{f}_{ji} = -\vec{f}_{ij}$$

Interatomic interaction: Lennard-Jones potential

Lennard-Jones potential:

$$U\left(r_{ij}\right) = 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

$$\frac{dU\left(r_{ij}\right)}{dr_{ij}} = 4\varepsilon\left[-12\frac{\sigma^{12}}{r_{ij}^{13}} + 6\frac{\sigma^{6}}{r_{ij}^{7}}\right]$$

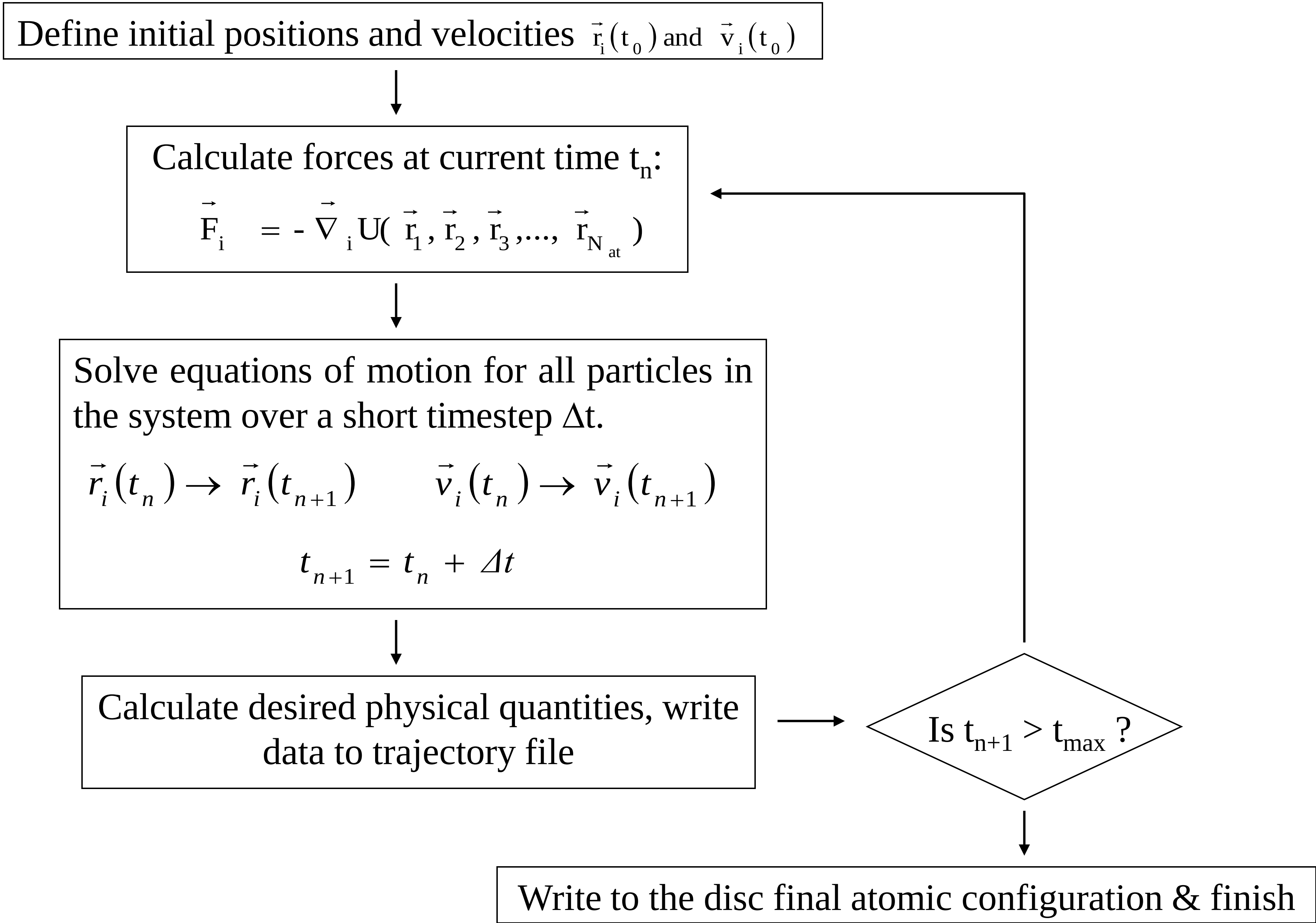


$$\vec{F}_i = -\sum_{j\neq i}\left(\widehat{x}\frac{x_{ij}}{r_{ij}} + \widehat{y}\frac{y_{ij}}{r_{ij}} + \widehat{z}\frac{z_{ij}}{r_{ij}}\right)\frac{dU(r_{ij})}{dr_{ij}} = -\sum_{j\neq i}\left(\frac{\vec{r}_i - \vec{r}_j}{r_{ij}}\right)\frac{dU(r_{ij})}{dr_{ij}}$$

$$\vec{F}_i = -\sum_{j\neq i} 24\frac{\varepsilon\sigma^{6}\left(\vec{r}_i - \vec{r}_j\right)}{r_{ij}^{8}}\left[1 - 2\left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

You will need to implement the calculation of Lennard-Jones potential in Homework #1

Schematic diagram of a basic MD code



Define initial positions and velocities $\vec{r}_i(t_0)$ and $\vec{v}_i(t_0)$

Calculate forces at current time t_n:

$$\vec{F}_i = -\vec{\nabla}_i U(\vec{r}_1, \vec{r}_2, \vec{r}_3, \ldots, \vec{r}_{N_{at}})$$

Solve equations of motion for all particles in the system over a short timestep Δt.

$$\vec{r}_i(t_n) \rightarrow \vec{r}_i(t_{n+1}) \qquad \vec{v}_i(t_n) \rightarrow \vec{v}_i(t_{n+1})$$

$$t_{n+1} = t_n + \Delta t$$

Calculate desired physical quantities, write data to trajectory file

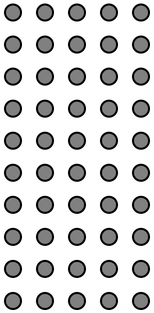
Is $t_{n+1} > t_{max}$?

Write to the disc final atomic configuration & finish

Homework #1: Write simple MD code with Velocity Verlet algorithm

(1) Write a simple **two-dimensional (2D)** “MD code” for particles interacting through Lennard-Jones potential. In the 2D system all atoms are moving on the same plane – you create the system by defining *x* and *y* coordinates of particle only and solve the equations of motion for *x* and *y* coordinates/velocities. Attach listing of your code to your homework report. You can use any of the languages you studied/used in other sections of CS6014.

(2) Create the initial system that consists of 50 atoms arranged into square 5×10 lattice as shown in the drawing in this slide. Choose the “lattice parameter” (distance between the nearest atoms) to correspond to the minimum of the Lennard-Jones potential (list the value in your report). You can assume that the initial velocities of all atoms are zero.



(3) Define the parameters of your model to be that of Ar (mass of 40 amu, parameters for Lennard-Jones potential: ε = 0.0103 eV and σ = 3.405 Å). Define timestep of integration, Δt=0.01ps. Use an appropriate unit conversion to ensure that all parameters of your system are in the same system of units, e.g. SI units.

(4) Perform MD simulation for the system you created in step (2) using the code you wrote. The total time of the simulation should be 50 ps or more. Make plots of the kinetic, potential, and total energies versus time for the duration of the simulations. Please show energies in eV and time in ps in your plots.

(5) Make plots of the initial and final atomic configuration that you obtained in the simulation. You can use your favorite plotting program to plot data points for x and y coordinates (do not connect points by lines). Or use “scatter plot” option if your graphic program has one. Or download one of the available free visualization packages for atomic structures. Please use Angstroms as units of coordinates in your plots.

(6) **Provide a physical discussion/explanation** of the final atomic structure and the time dependences of the total, potential, and kinetic energies. Try to be quantitative in your discussion of the values of energies.

Your code can have the following structure:

Define the parameters of the system (mass of an atom m), initial coordinates and velocities of the particles in the system (x_i^0, y_i^0, $v_i^{x,0}$, $v_i^{y,0}$), set initial time $t^0 = 0$, total time of the simulation, timestep of integration, Δt.

Calculate the two components of the initial force $F_i^{x,init}$, $F_i^{y,init}$ for each particle *i*

$F_i^{x,old} = F_i^{x,init}$, $F_i^{y,old} = F_i^{y,init}$, $x_i^{old}= x_i^0$, $y_i^{old}= y_i^0$, $v_i^{x,old}= v_i^{x,0}$, $v_i^{y,old}= v_i^{y,0}$ **- use a subroutine for calculation of forces**

```
Time loop:
    Loop over all atoms:
        1. Calculate position at the current time
           $x_i^{new} = x_i^{old} + \Delta t\, v_i^{x,old} + \Delta t^2 F_i^{x,old}/2m$      $y_i^{new} = y_i^{old} + \Delta t\, v_i^{y,old} + \Delta t^2 F_i^{y,old}/2m$
    End loop for atoms

        2. Calculate new forces, $F_i^{x,new}$, $F_i^{y,new}$ (and potential energy of the system) – use a subroutine

    Loop over all atoms:
        3. Calculate new velocities (and kinetic energy of the system)
           $v_i^{x,new} = v_i^{x,old} + \Delta t(F_i^{x,old} + F_i^{x,new})/2m$      $v_i^{y,new} = v_i^{y,old} + \Delta t(F_i^{y,old} + F_i^{y,new})/2m$

        4. Copy new variables to old ones (actually, you do not need to store the values of coordinates and velocities at
           two different times, you can use the same variables and write new values over the old ones at steps 1 and 3).

    End loop for atoms

    5. Write potential, kinetic, and total energies of the system to disc for further analysis.

    6. Update time ($t_n = t_{n-1} + \Delta t$). If $t_n$ is less than the time of the simulation, continue with the time loop
End of the time loop, write the final atomic configuration to the disc and stop of simulation
```

Note that the potential energy given by Lennard-Jones interatomic potential is for a pair of atoms (*i* and *j*) – you should divide it between the two atoms.