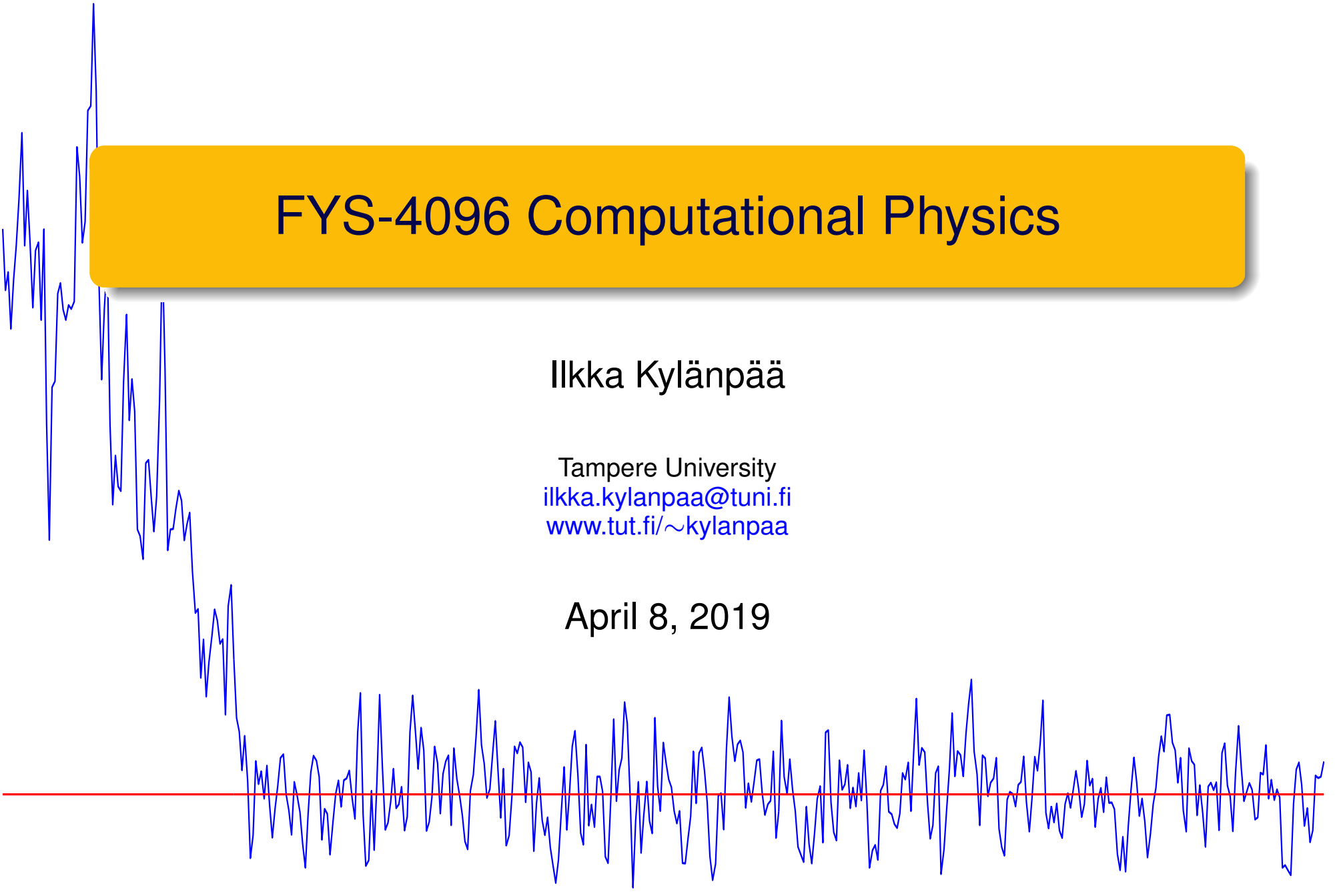
# FYS-4096 Computational Physics

Ilkka Kylänpää

Tampere University
ilkka.kylanpaa@tuni.fi
www.tut.fi/∼kylanpaa

April 8, 2019



# Course details

Active participation for 12 weeks that include

- $\sim 2$ hours of lectures on Mondays (12-14)
- $3 - 4$ hours in computer laboratory on Tuesdays (14-18)
- $\sim 2$ hours of exercises on Fridays (12-14)

Course requirements and practicalities:

- Good participation and mandatory weekly exercises (70% of the grade)
- Two project works (30% of the grade)
- However, both project works required to pass the course.
- Weekly exercises are to be presented and discussed on Friday, that is, prepare a few slides demonstrating your solutions.
  - **2 presentations and minimum 4 attendances at Friday's exercise sessions are required to pass the course**

# Timetable (in calendar weeks)

Third period

| Week | Description |
|---|---|
| 2 | Course details, Linux, Git, Python, derivatives and integration |
| 3 | Interpolation, root search, steepest descent |
| 4 | Gaussian elimination, LU decomposition, conjugate gradient, eigenvalue equations and Krylov subspace |
| 5 | Periodic boundary conditions, reciprocal lattice and Wigner-Seitz cell, crystal structure and electron densities |
| 6-7 | Solving differential equations: finite difference and finite element method |
| 8-9 | Project work I and exam week (No teaching or exercises) |

Fourth period

| | |
|---|---|
| 10-15 | Methods and applications |
| 16-17 | Project work II and exam week (No teaching or exercises) |

# Timetable (in calendar weeks)

Third period

| Week | Description |
|---|---|
| 2-7 | Getting familiar with general tools in computational physics |
| 8-9 | Project work I and exam week (No teaching or exercises) |

Fourth period (Methods and applications)

| | |
|---|---|
| 10 | Many-body Schrödinger equation, Born–Oppenheimer approximation, Hartree approximation |
| 11 | Hartree-Fock and Density Functional Theory |
| 12 | Variational and Diffusion Monte Carlo |
| 13 | Path integral Monte Carlo |
| 14 | Classical Monte Carlo |
| 15 | Molecular Dynamics (and Car-Parinello Molecular Dynamics) |
| 16-17 | Project work II and exam week (No teaching or exercises) |

# Molecular dynamics

The previous Monte Carlo methods have proven to be rather efficient and straigtforward means for studying equilibrium properties of many body systems. However, with the previous Monte Carlo approaches we are not able to address, for example, how fast (in real time) the system will find equilibrium after a sudden change in the temperature.

This kind of real time information can be straightforwardly accessed via molecular dynamics (MD) simulations. In MD simulations, for example, the real time dynamics of a set of atoms is modeled by solving classical equations of motion – through Lagrangian mechanics or Newtonian mechanics.

The atoms can form molecules, clusters, solids, etc., and the whole system can be macroscopically in one or many phases such as solid, liquid or gas. Therefore, MD simulations can be used in modeling, e.g., atomic level real time events.

# Molecular dynamics

For a particle (e.g., atom, molecule or a cluster) with mass $m_i$ the Newton equation of motion is given as

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i, \tag{1}$$

where $\mathbf{F}_i$ is the net force acting on particle $i$.

That is, the classical real time dynamics of a single particle under the influence of all other particles reduces to solving ordinary differential equation

$$\begin{bmatrix} \frac{d\mathbf{v}_i}{dt} \\ \frac{d\mathbf{r}_i}{dt} \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{F}_i}{m_i} \\ \mathbf{v}_i \end{bmatrix}, \tag{2}$$

for all $i$.

# Molecular dynamics

For conservative forces $\mathbf{F} = -\nabla V$. Therefore, if we have knowledge of the interaction potential between the constituents in the simulation, we can obtain the needed force by calculating the negative gradient of the potential energy.

**The more accurate the (approximate) interaction potentials, the more realistic and predictive simulations.**

For central potentials, i.e., the interaction potential is only dependent on the distance $r$ between the two interacting particles ($V = V(r)$), the force can be expressed as

$$\mathbf{F} = -\nabla V(r) = -\frac{\partial V}{\partial r}\hat{\mathbf{r}}, \tag{3}$$

where $\hat{\mathbf{r}}$ is a unit vector in the direction of the force.

# Molecular dynamics

For $N$ particles interacting via two body potentials the net force on particle $i$ can be expressed as

$$\begin{aligned}
\mathbf{F}_i &= -\nabla_i U(R) \\
&= -\nabla_i \sum_{m>n} V(r_{mn}) \\
&= -\nabla_i \sum_{m\neq i} V(r_{mi}) \\
&= \sum_{m\neq i} -\nabla_i V(r_{mi}) \\
&= \sum_{m\neq i} \mathbf{F}_{mi}.
\end{aligned} \tag{4}$$

Notice that since $\mathbf{F}_{mi} = -\mathbf{F}_{im}$ all the pair terms need only be calculated once.

# Molecular dynamics

Once you know how to compute the forces you can proceed, in principle, the same way as before with ordinary differential equations.

However, once the degrees of freedom in your system increases you might find, e.g., the considered Runge-Kutta methods to be too inefficient.

This inefficiency is due to the fact that, for example, in the fourth order Runge-Kutta you would need to calculate the forces 4 times. But calculation of forces is the most time consuming part of the simulation, which is why you would like to do that as few times as possible.

(Additionally, Runge-Kutta lacks time-reversal symmetry of Newton's equations.)

**Leap frog method:**

$$\mathbf{v}_{n+1} = \mathbf{v}_{n-1} + 2\Delta t \frac{1}{m}\mathbf{F}_n, \tag{5}$$

$$\mathbf{r}_{n+2} = \mathbf{r}_n + 2\Delta t \mathbf{v}_{n+1} \tag{6}$$

Needs $\mathbf{v}_{-1}$ in addition to initial conditions, e.g., using backward Euler.

**Verlet method:**

$$\mathbf{v}_n = \frac{\mathbf{r}_{n+1} - \mathbf{r}_{n-1}}{2\Delta t} \tag{7}$$

$$\mathbf{r}_{n+1} = 2\mathbf{r}_n - \mathbf{r}_{n-1} + \Delta t^2 \frac{1}{m}\mathbf{F}_n \tag{8}$$

Needs $\mathbf{r}_{-1}$ in addition to initial conditions, e.g., from

$$\mathbf{r}_{-1} = \mathbf{r}_0 - \Delta t \mathbf{v}_0 + \Delta t^2 \frac{1}{2m}\mathbf{F}_0. \tag{9}$$

**Velocity Verlet:**

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \Delta t \mathbf{v}_n + \Delta t^2 \frac{1}{2m} \mathbf{F}_n, \tag{10}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t \frac{1}{2m} \left( \mathbf{F}_{n+1} + \mathbf{F}_n \right). \tag{11}$$

See more details and algorithms from the ordinary differential part of the course.

# Molecular dynamics

**Initial configuration in atomistic simulations:**

For a liquid or gas you could in principle place the atoms / molecules randomly inside the simulation cell. However, this could lead to substantial overlaps in configurations, which in MD can cause computational problems.

Therefore, starting from a lattice is the usual choise also for liquids and gases. For example, the face-centered cubic (FCC) lattice.

During the simulation the lattice structure will disappear. This “melting” process can be enhanced by giving small random displacements for the atoms / molecules.

In case of molecular fluid also the initial orientation of the molecules might lead to computational challenges (due to the overlap), especially at high densities.

**Initial velocities:**

For MD simulations initial velocities of all the atoms or molecules must be specified.

Random velocities from a Gaussian (Maxwell-Boltzmann) distribution

$$f(v_{ix}) = \left(\frac{m_i}{2\pi k_{\mathrm{B}} T}\right)^{1/2} \exp\left(\frac{m_i v_{ix}^2}{2k_{\mathrm{B}} T}\right) \tag{12}$$

is a common choice and conforms to a specific temperature. These can be further corrected for zero net momentum, i.e.,

$$\sum_{i=1}^{N} m_i \mathbf{v}_i = \mathbf{0}. \tag{13}$$

An alternative way, needing longer equilibration, is simply to choose each velocity component randomly from uniform distribution in range $(-v_{\max}, v_{\max})$.

**Periodic boundary conditions (PBC):**

Considering liquids, gases, and solids, the periodic boundary conditions are required.

For a general simulation cell, you can set your simulation cell vectors as your "lattice vectors" and use the expression from previous lectures, i.e., $\mathbf{A}\alpha = \mathbf{r}$, to always set $\alpha$ modulo 1. This ensures that the simulated particles are kept inside the simulation cell.

The **minimum image convention** refers to measuring interparticle distances in periodic simulations. For example, let's consider measuring the distance from $\mathbf{r}_1$ to $\mathbf{r}_2$ in 3D. The basic idea is to choose that specific image of $\mathbf{r}_2$, i.e. $\mathbf{r}_2 + n_1\mathbf{a}_1 + n_2\mathbf{a}_2 + n_3\mathbf{a}_3$, that is closest to $\mathbf{r}_1$.

The minimum image distance is thus obtained by considering components of $\Delta\alpha = \mathrm{modulo}(A^{-1}(\mathbf{r}_2 - \mathbf{r}_1), \mathbf{1})$. If $|\Delta\alpha_i| > 0.5$, then that component needs to be modified as $\Delta\alpha_i = \Delta\alpha_i - 1$ if $\Delta\alpha_i > 0.5$, and as $\Delta\alpha_i = \Delta\alpha_i + 1$ if $\Delta\alpha_i < -0.5$.

# Molecular dynamics

For so-called short range potentials this is close to all you need to know for coding and performing periodic MD simulations. That is, using for example Lennard-Jones type potentials for interactions between atoms.

In case of long range potentials, such as the Coulomb potential, you need to perform summation over images for obtaining the periodic potential. For example, for the 3D Coulomb in atomic units this would read

$$V(R) = \frac{1}{2}\sum_{\mathbf{n}}{}' \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{Z_i Z_j}{\|\mathbf{r}_{ij} + n_1\mathbf{a}_1 + n_2\mathbf{a}_2 + n_3\mathbf{a}_3\|}, \tag{14}$$

where the prime in the sum indicates that for $\mathbf{n} = \mathbf{0}$ we will enforce $i \neq j$.

A long range interaction is often as one in which the spatial interaction falls off a $r^{-a}$, where $a \leq d$ and $d$ is the dimensionality of the system. For example, dipole-dipole interaction goes as $r^{-3}$ in 3D and the Coulomb as $r^{-1}$.

# Molecular dynamics

The sum over periodic images (Madelung sum)

$$V(R) = \frac{1}{2} \sum_{\mathbf{n}}{}' \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{Z_i Z_j}{\|\mathbf{r}_{ij} + n_1\mathbf{a}_1 + n_2\mathbf{a}_2 + n_3\mathbf{a}_3\|} \tag{15}$$

is only conditionally convergent, and thus, it is conventionally performed as a so-called Ewald sum.

In the Ewald sum the point charges are screened by Gaussian charge distributions, and the screening is balanced by cancelling distributions. The sum over images for the screened point charges converges well in real space, and the cancelling distribution converges well in reciprocal space.

Practical way for performing the Ewald sum is so-called particle mesh Ewald. Improvements over single Gaussian Ewald sum are, for example, a so-called extended Ewald summation technique and especially the "optimized method for treating long range potentials".

# Molecular dynamics: observables

Energy is naturally one desired observable from simulations. In MD the used algorithm should also be tested, for example, by considering the conservation of total energy. One way for doing this is looking at the root-mean-square value of the energy, i.e.,

$$E_{\mathrm{RMS}} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}[E(0) - E(i\Delta t)]^2}. \tag{16}$$

This can be used in observing a drift in the total energy (which is not a desired property). Conservation of the total momentum can also become necessary.

The energy at equilibrium is simply the average of the kinetic energy + the potential energy, i.e.,

$$\langle E \rangle = \langle K + V \rangle = \left\langle \sum_{i=1}^{N} \frac{1}{2} m_i v_i^2 + V(R) \right\rangle. \tag{17}$$

# Molecular dynamics: observables

Notice that via the equipartition theorem the temperature of the system can be obtained from

$$\frac{dN}{2} k_{\mathrm{B}} T = \langle K \rangle, \tag{18}$$

where $d$ is the dimensionality.

Pair correlation function $g(r)$ describes the (conditional) probability of finding two particles at distance $r$ apart from each other. For example, in case of monoatomic fluid it can be expressed as

$$4\pi r^2 g(r) = \frac{V}{N^2} \left\langle \sum_{i=1}^{N} \sum_{i \neq j} \delta(r - r_{ij}) \right\rangle, \tag{19}$$

where $V$ is the volume of the simulation cell and $N$ is the number of particles.