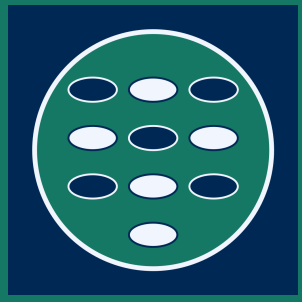
Noregs miljø- og biovitskaplege universitet

Institutt for datavitskap



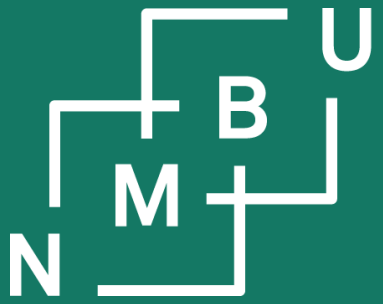
Digitalisering på Ås

INF203

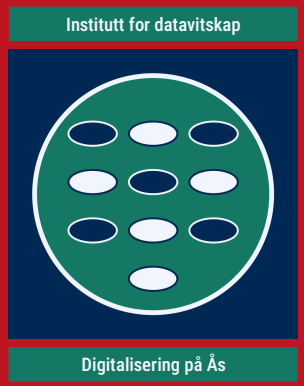
June advanced programming project

1 Introduction

1.1 Course structure
1.2 June problem intro
1.3 Group management
1.4 Agile methods
1.5 Requirements analysis
1.6 Good documentation



Noregs miljø- og biovitskaplege universitet



1 Introduction

1.1 Course structure

1.2 Programming task intro

1.3 Group management

1.4 Agile methods

INF203

3rd June 2025

Overarching idea

Main objective: Learn and practice the process of developing code effectively.

- Understand a numerical problem from scientific practice; be able to analyse it on paper. Analyse and address it as a practical task.
- Design software and proceed to implementation based on a clear plan.
- Debug, test, validate, document, and present your implementation.

Some elements of the programming process that we will consider:

- Agile methods for requirements analysis and development
- Developing software as a team, using a shared repository
- Good practices for designing object oriented code, based on co-design of data and software
- Implementing and documenting good object oriented code in Python

Estimated course structure

In this course, it is mainly you who do the work, as it is a practice course.

There will also be contributed lectures, and we have scheduled discussions.

- The course INF203 has the same status as INF202, but is not identical.
- Every year, in INF203 we will be working on a new problem, which hopefully is of interest to people and also connected to our research.
- It is a living course, where our discussions develop at real time.

That said, there are four main parts of the course, each with one submission:

- Introduction (leading to worksheet 1, deadline 5th June, 17.30)
- Object orientation in practice (… worksheet 2, d'line 11th June, 17.30)
- Debugging and validation (… worksheet 3, deadline 16th June, 17.30)
- Development for production (… final project, d'line 19th June, 17.30)

Plan for part 1: Introduction

Tuesday, 3rd June 2025

- Part 1.2: June problem intro
  Introducing the aim: Implement MC simulation with test-area method for the surface tension of the Lennard-Jones truncated-shifted fluid.
- Part 1.3: Group management
- Part 1.4: Agile methods

Wednesday, 4th June 2025

- Part 1.5: Requirements analysis
- Part 1.6: Good documentation

Group submission of the first worksheet by Thursday, 5th June, 17.30 CEST.

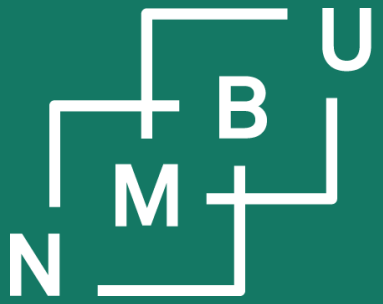
Overarching idea

Main objective: Learn and practice the process of developing code effectively.

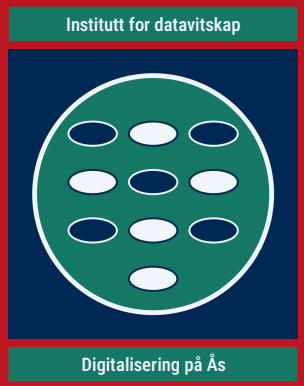
- Understand a numerical problem from scientific practice; be able to analyse it on paper. Analyse and address it as a practical task.
- Design software and proceed to implementation based on a clear plan.
- Debug, test, validate, document, and present your implementation.

Some elements of the programming process that we will consider:

- Agile methods for requirements analysis and development
- Developing software as a team, using a shared repository
- Good practices for designing object oriented code, based on co-design of data and software
- Implementing and documenting good object oriented code in Python



Noregs miljø- og biovitskaplege universitet



1 Introduction

1.1 Course structure

1.2 Programming task intro

1.3 Group management

1.4 Agile methods

INF203

3rd June 2025

Disclaimer

Depending on what background from physics or engineering you have, you may or may not understand the Sampayo paper.[2]

Or even the Gloor paper,[1] which is the original source for the test area method.

This is *not* needed! It is *not* expected!

The method is actually simple, if explained, for which we have good time.

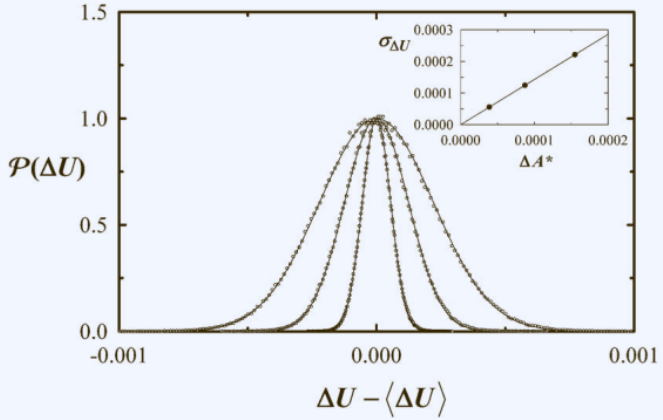


FIG. 1. TA deformations of a planar liquid-vapor interface of the LJ-TS fluid. MD simulations of $N=749$ particles in a periodic box of dimensions $L_x=L_y=7.885\sigma$ and $Lz=6L_x$ at $T^*=kT/\epsilon=0.8$ over 3×10^6 timesteps. The deformations correspond to changes in the box dimensions (particle coordinates) of $L'_x=L_x\sqrt{1+\xi}$, $L'_y=L_y\sqrt{1+\xi}$, and $L'_z=L_z/(1+\xi)$. The distribution $\mathcal{P}(\Delta U)$ of the change in energy (relative to its average in units of ϵ) scaled at the maximum peak height for different relative deformations ΔA^*. The width (standard deviation, $\sigma_{\Delta U}$) is depicted in the inset.

The tension is obtained as the change in free energy per unit area for infinitesimal perturbations to $\mathcal{O}(\langle \Delta U^3 \rangle)$,

$$\gamma = \lim_{\Delta A \to 0} \frac{\Delta F}{\Delta A} = \lim_{\Delta A \to 0} \left\{ \frac{\Delta F_1}{\Delta A} + \frac{\Delta F_2}{\Delta A} + \frac{\Delta F_3}{\Delta A} \right\}. \qquad (5)$$

[1] G. J. Gloor *et al.*, *Journal of Chemical Physics* **123**: 134703, doi:10.1063/1.2038827, **2005**.

[2] J. S. Sampayo *et al.*, *Journal of Chemical Physics* **132**: 141101, doi:10.1063/1.3376612, **2010**.

Specific problem: Test area method[1, 2]

the change in the Helmholtz free energy F is expressed thermodynamically as[1]

$$dF = -SdT - p_g dV_g + p_l dV_l + \mu dN + \gamma dA + CdR, \qquad (1)$$

where S is the entropy, $V_{g,l}$ are the vapor and liquid volumes, T is the temperature, μ is the chemical potential, N is the number of particles, A is the interface area, and C is the conjugate variable for R. The surface tension of a drop is given by

$$\left(\frac{\partial F}{\partial A}\right)_{NVT} = \gamma_s, \qquad (2)$$

where the minimal interfacial tension γ_s defines R_s and corresponds to taking $C=0$. The change in free energy ΔF due to a virtual change in area ΔA can be expressed as the average of the Boltzmann factor of the corresponding change in configurational energy ΔU,[24]

$$\Delta F = -kT \ln\left\langle \exp\left(-\frac{\Delta U}{kT}\right)\right\rangle \qquad (3)$$

$$= \langle \Delta U\rangle - \frac{1}{2kT}\{\langle \Delta U^2\rangle - \langle \Delta U\rangle^2\}$$

$$+ \frac{1}{6(kT)^2}\{\langle \Delta U^3\rangle - 3\langle \Delta U^2\rangle\langle \Delta U\rangle + 2\langle \Delta U\rangle^3\}. \qquad (4)$$

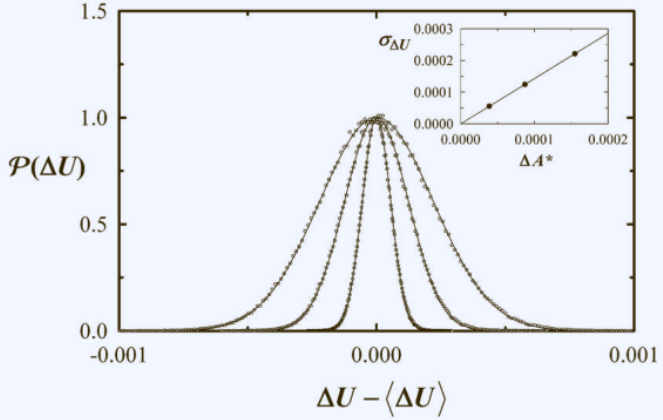


FIG. 1. TA deformations of a planar liquid-vapor interface of the LJ-TS fluid. MD simulations of $N=749$ particles in a periodic box of dimensions $L_x=L_y=7.885\sigma$ and $Lz=6L_x$ at $T^*=kT/\epsilon=0.8$ over 3×10^6 timesteps. The deformations correspond to changes in the box dimensions (particle coordinates) of $L'_x=L_x\sqrt{1+\xi}$, $L'_y=L_y\sqrt{1+\xi}$, and $L'_z=L_z/(1+\xi)$. The distribution $\mathcal{P}(\Delta U)$ of the change in energy (relative to its average in units of ϵ) scaled at the maximum peak height for different relative deformations ΔA^*. The width (standard deviation, $\sigma_{\Delta U}$) is depicted in the inset.

The tension is obtained as the change in free energy per unit area for infinitesimal perturbations to $\mathcal{O}(\langle \Delta U^3\rangle)$,

$$\gamma = \lim_{\Delta A\to 0}\frac{\Delta F}{\Delta A} = \lim_{\Delta A\to 0}\left\{\frac{\Delta F_1}{\Delta A} + \frac{\Delta F_2}{\Delta A} + \frac{\Delta F_3}{\Delta A}\right\}. \qquad (5)$$

[1]G. J. Gloor *et al.*, *Journal of Chemical Physics* **123**: 134703, doi:10.1063/1.2038827, **2005**.

[2]J. S. Sampayo *et al.*, *Journal of Chemical Physics* **132**: 141101, doi:10.1063/1.3376612, **2010**.

How to get started?

Let us have a look at the first worksheet:

1) Group collaboration

2) Start coding: Molecules, box, and random configuration generation

3) Implementation of the LJTS potential

4) Requirements register
   - We will discuss the methodology for this tomorrow.
   - It is based on formulating *epics* and *user stories* for *personas*.

5) Set up a LaTeX document
   - You need to find a viable way of editing/managing it as a group.

Molecular model: Lennard-Jones truncated-shifted

The potential energy is given as a sum over all pairs of molecules

$$E_{\text{pot}} = \Sigma_i \Sigma_{j>i} u(r_{ij}).$$

Therein, r_{ij} is the distance between molecules i and j. The truncated-shifted Lennard-Jones potential $u(r_{ij})$ is given by:

$u(r_{ij}) = u_{\text{LJ}}(r_{ij}) - u_{\text{LJ}}(2.5)$, if $r_{ij} < 2.5$
$u(r_{ij}) = 0$, if $r_{ij} \geq 2.5$

There, u_{LJ} is the Lennard-Jones potential, given by $u_{\text{LJ}}(r) = 4 \cdot (1/r^{12} - 1/r^6)$.

So the Lennard-Jones potential is being truncated at a cutoff distance of 2.5 and shifted by adding $-u_{\text{LJ}}(2.5) = +0.0163169$. All pairs of molecules i and j with a distance below 2.5 contribute to the potential energy.

Cycles with increased code functionality

The recommendation is that you follow the worksheets.

- The first worksheet leads, minimally, to a basic implementation of the **LJTS potential** model for a system with periodic boundary conditions.
- The second worksheet leads, minimally, to a functioning **Monte Carlo simulation** code.
- The third worksheet leads, minimally, to an implementation of the **test area method** for the surface tension.
- The fourth worksheet leads toward the mandatory **complete project** submission, which minimally also includes a report and some results.

Only the final submission is mandatory.
You can in principle ignore the worksheets completely and still get an A.

Monte Carlo using the Metropolis algorithm

Problem: We want to sample high-dimensional configurations **q**.

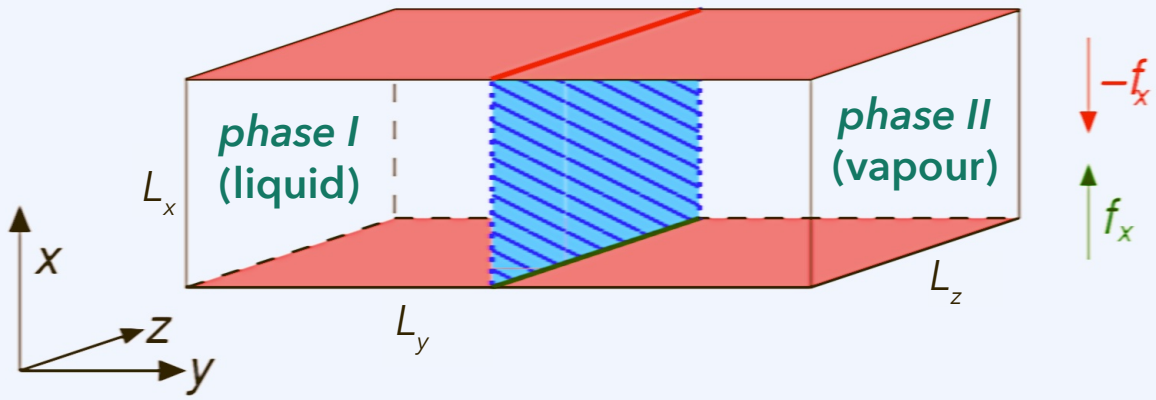
There is a probability distribution $\rho(\mathbf{q})$. That distribution is also known as the phase space density. For our problem, it is given by $\rho(\mathbf{q}) = \exp(-E_{\text{pot}}(\mathbf{q}) / T)$.

Metropolis algorithm: Iteratively proceed as follows.

1) Select a random molecule and move it a bit at random.
2) Compute how much the potential energy has changed.
   i. If the potential energy has decreased, always accept the move.
   ii. If it has increased, let $\Delta E_{\text{pot}} > 0$ be that increase. Then accept the move with the probability $\exp(-\Delta E_{\text{pot}} / T)$, which is between 0 and 1.
3) The present configuration now counts for the sampling. If it is unchanged from the previous one, it also counts (*i.e.* it counts again).

By a common convention, if you have N molecules, "one MC loop" (or one step in the MC simulation) consists of executing the loop above for N times.

Pressure tensor at a planar interface



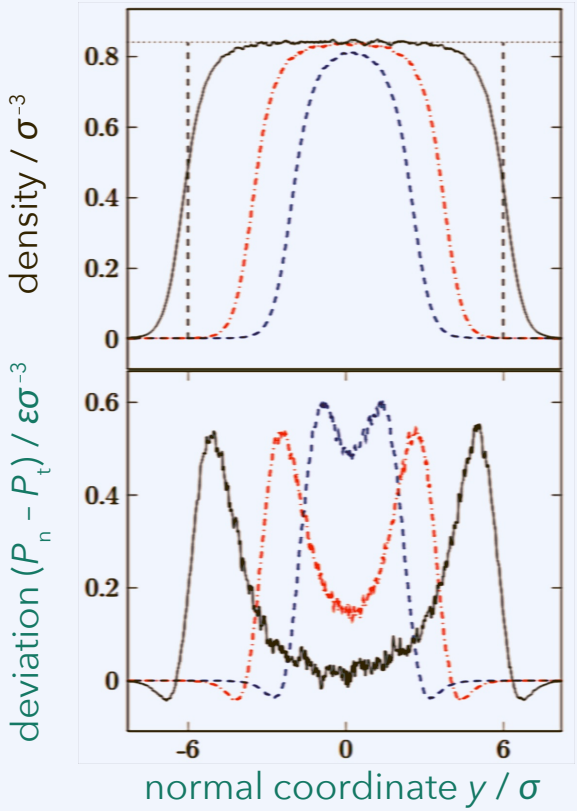
- Mechanical equilibrium → fluid at rest, off-diagonal entries in pressure tensor are zero.
- The tangential pressure $P_t = P^{xx} = P^{zz}$ deviates from the normal pressure $P_n = P^{yy}$.
- Compared to the system boundaries perpendicular to *y* direction (normal direction), those in tangential direction experience an additional tangential force inward:
  - In *x* direction, $f_x \sim L_z$; similarly, in z direction, $f_z \sim L_x$.
  - By symmetry, the proportionality factors are equal, $f_x = -\gamma L_z$ and $f_z = -\gamma L_x$.
- For *x* direction, $P^{xx} - P_n = f_x / L_y L_z = -\gamma L_z / L_y L_z = -\gamma / L_y$. Therefore, $\gamma = L_y (P_n - P^{xx})$.
- The same argument applies in *z* direction. So we can write $\gamma = L_y (P_n - P_t)$.

From macro- to micromechanics

Using localized **pressure profiles** based on **continuum mechanics**:

$$\gamma = \int_y dy \left(P_n(y) - P_t(y) \right)$$

vapour-liquid surface tension from integral over deviation between P_n and P_t



LJ fluid

$T = 0.7\ \varepsilon$

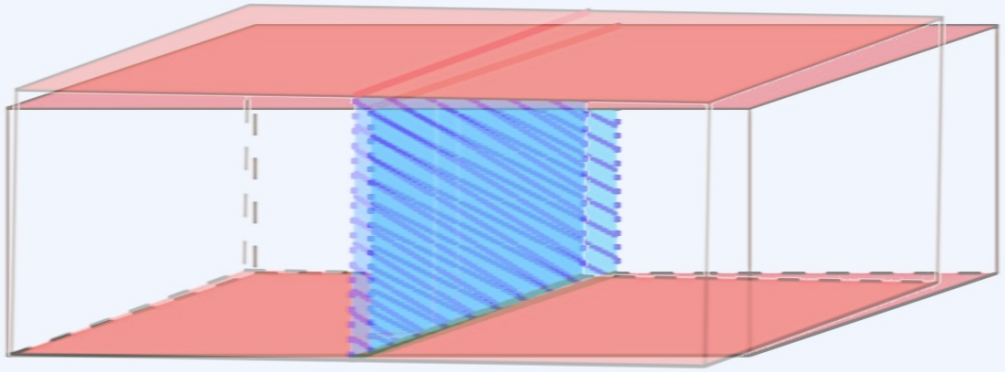
from S. Werth *et al.*, *Phys. A* 392: 2359, **2013**.

Compare the expression using the pressure P_n and P_t for the whole system:

- For x direction, $P^{xx} - P_n = f_x / L_y L_z = -\gamma L_z / L_y L_z = -\gamma / L_y$. Therefore, $\gamma = L_y (P_n - P^{xx})$.
- The same argument applies in z direction. So we can write $\gamma = L_y (P_n - P_t)$.

Grounding in statistical mechanics

One way of obtaining a statistical mechanical expression for the surface tension is through a hypothetical **infinitesimal test transformation:**



canonical ensemble

- **N**, V and T constant
- V' and V'' (liq, vap) constant
- Surface area a changes

$$\gamma = \left(\frac{\partial F}{\partial a}\right)_{\mathbf{N},V',V'',T}$$

For a system of point masses, this leads to the **same result as the argument from mechanics**, $\gamma = a^{-1}(\langle\Pi_n\rangle - \langle\Pi_t\rangle)$; there is no contribution from ρT.

Pressure obtained as $P_n = \rho T + V^{-1}\langle\Pi_n\rangle$ and $P_t = \rho T + V^{-1}\langle\Pi_t\rangle$.

- Therein, Π is the virial, $\Pi_n = \Pi^{yy} = \Sigma_{\{i,j\}} r_{ij,y} f_{ij,y}$ and $\Pi_t = \Pi^{xx} = \Sigma_{\{i,j\}} r_{ij,x} f_{ij,x}$.
- Since $V = aL_y$, where a is the surface area, $\gamma = a^{-1}(\langle\Pi_n\rangle - \langle\Pi_t\rangle)$. **Note that ρT cancels out!**
- While the virial for the whole system is uniquely defined, the local profiles are not.

Second look at the Sampayo paper[1]

the change in the Helmholtz free energy F is expressed thermodynamically as[1]

$$dF = -SdT - p_g dV_g + p_l dV_l + \mu dN + \gamma dA + CdR, \qquad (1)$$

where S is the entropy, $V_{g,l}$ are the vapor and liquid volumes, T is the temperature, μ is the chemical potential, N is the number of particles, A is the interface area, and C is the conjugate variable for R. The surface tension of a drop is given by

$$\left(\frac{\partial F}{\partial A}\right)_{NVT} = \gamma_s, \qquad (2)$$

where the minimal interfacial tension γ_s defines R_s and corresponds to taking $C=0$. The change in free energy ΔF due to a virtual change in area ΔA can be expressed as the average of the Boltzmann factor of the corresponding change in configurational energy ΔU,[24]

$$\Delta F = -kT \ln\left\langle \exp\left(-\frac{\Delta U}{kT}\right)\right\rangle \qquad (3)$$

$$\Delta F = \underbrace{\langle \Delta U\rangle}_{\Delta F_1} \underbrace{-\frac{1}{2kT}\{\langle \Delta U^2\rangle - \langle \Delta U\rangle^2\}}_{\Delta F_2} + \underbrace{\frac{1}{6(kT)^2}\{\langle \Delta U^3\rangle - 3\langle \Delta U^2\rangle\langle \Delta U\rangle + 2\langle \Delta U\rangle^3\}}_{\Delta F_3}. \qquad (4)$$

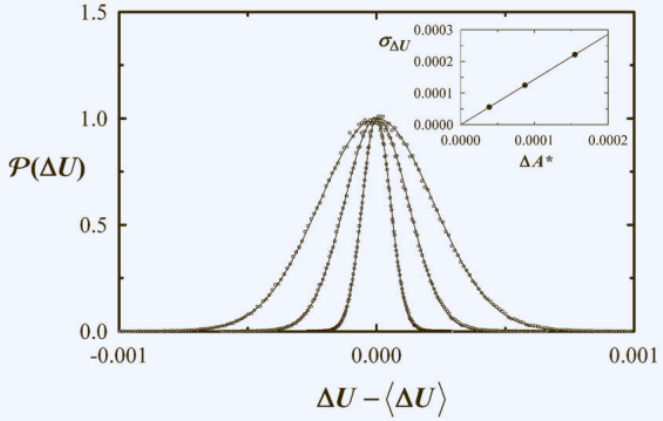
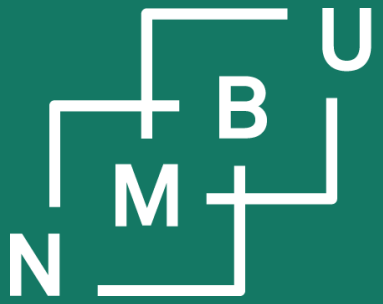


FIG. 1. TA deformations of a planar liquid-vapor interface of the LJ-TS fluid. MD simulations of $N=749$ particles in a periodic box of dimensions $L_x=L_y=7.885\sigma$ and $Lz=6L_x$ at $T^*=kT/\epsilon=0.8$ over 3×10^6 timesteps. The deformations correspond to changes in the box dimensions (particle coordinates) of $L_x'=L_x\sqrt{1+\xi}$, $L_y'=L_y\sqrt{1+\xi}$, and $L_z'=L_z/(1+\xi)$. The distribution $\mathcal{P}(\Delta U)$ of the change in energy (relative to its average in units of ϵ) scaled at the maximum peak height for different relative deformations ΔA^*. The width (standard deviation, $\sigma_{\Delta U}$) is depicted in the inset.

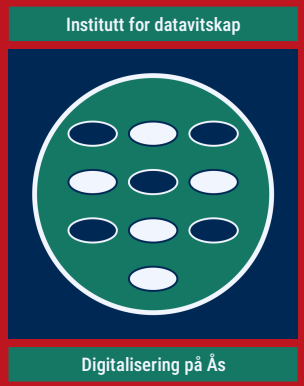
The tension is obtained as the change in free energy per unit area for infinitesimal perturbations to $\mathcal{O}(\langle \Delta U^3\rangle)$,

$$\gamma = \lim_{\Delta A\to 0}\frac{\Delta F}{\Delta A} = \lim_{\Delta A\to 0}\left\{\frac{\Delta F_1}{\Delta A} + \frac{\Delta F_2}{\Delta A} + \frac{\Delta F_3}{\Delta A}\right\}. \qquad (5)$$

[1]J. S. Sampayo *et al.*, *Journal of Chemical Physics* **132**: 141101, doi:10.1063/1.3376612, **2010**.



Noregs miljø- og biovitskaplege universitet



1 Introduction

1.1 Course structure

1.2 Programming task intro

1.3 Group management

1.4 Agile methods

INF203

3rd June 2025

Participation expected from groups

Norwegian University of Life Sciences

Overview over the structured activities and submissions for the groups.

- Each group **should** come to an informal meeting ("chat") – recommended, but not required.
- Each group **must** be present and give a 10-to-15 min. presentation at one of the "student presentations" sessions; this is obligatory.
- Each group **should** do the worksheet 1, 2, and 3 submissions – recommended, but not required.
- Each group **must** do the final submission of the complete project work – this will contribute 70% of the grade.
- Each group **must** do a colloquium in the week after the teaching period – this will contribute 30% of the grade.

By the date of this meeting, everybody who is actually participating in the course should be in one group on Canvas.

Booking of time slots

Norwegian University of Life Sciences

As a group, you can book slots for the chats, presentations, and colloquia.

- URL for booking time slots for informal discussions (chat) and group presentations related to one of the worksheets:

  https://terminplaner6.dfn.de/b/f707b6ca4dd47176783b855748eeb067-1244405

- URL for booking a time slot for the colloquium, which is like an oral exam for the group and will contribute 30% of the total grade:

  https://terminplaner6.dfn.de/b/d9a4b05377b4a4f1ef4ed326beb97905-1244450

Do not have multiple members of your group do a booking. Only one!

The booking forms close at the end of the discussion session on 4th June 2025.

Basic expectations

Be responsible. You are not alone; you have group partners who rely on you.

Your software is *your* task. You can discuss with other groups, but do not share the code. This is important for the presentations and other discussions that we will have as a group. So for example do not show your code when presenting.

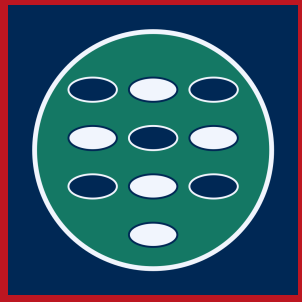
- If you have a very specific – unusual – code/solution detail that is identical with others, it causes suspicion, not just against them, but also you.
- It is also representative of actual development practice, where there often are restrictions as to what you are allowed to disclose to others.

All in the group must understand the entire code and be able to explain it. There will be a colloquium (*i.e.* oral exam) where we will check this.

Everybody must have the lead responsibility for something in the code. You will need to document this as part of your report.

Noregs miljø- og biovitskaplege universitet

Institutt for datavitskap



Digitalisering på Ås

1 Introduction

1.1 Course structure
1.2 Programming task intro
1.3 Group management
1.4 Agile methods

INF203

3rd June 2025

Agile manifesto[1]

The 12 principles of agile

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

[1]https://www.agilealliance.org/wp-content/uploads/2019/09/agile-manifesto-download-2019.pdf

Agile manifesto[1]

The 12 principles of agile

7 Working software is the primary measure of progress.

8 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9 Continuous attention to technical excellence and good design enhances agility.

10 Simplicity – the art of maximizing the amount of work not done – is essential.

11 The best architectures, requirements, and designs emerge from self-organizing teams.

12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

[1]https://www.agilealliance.org/wp-content/uploads/2019/09/agile-manifesto-download-2019.pdf

The traditional paradigm: “Waterfall”

In project management and code development, we can proceed sequentially:

Funding acquisition
Requirements analysis
Architecture design
Development / implementation
Testing and debugging
Deployment/delivery

This is called the “waterfall” paradigm.

Many claim it is not modern. Often the same people who claim this, nonetheless, require others to practice it, due to its main advantage: Control.

Some entity can specify in advance exactly what will happen.

The devil's paradigm: Worst of both worlds

Some managers argue: You must be agile! It is the modern way.

At the same time, they like to control things, plan details in advance, and follow up on what they asked for. They love it because it gives them power.

This creates the devil's paradigm where:

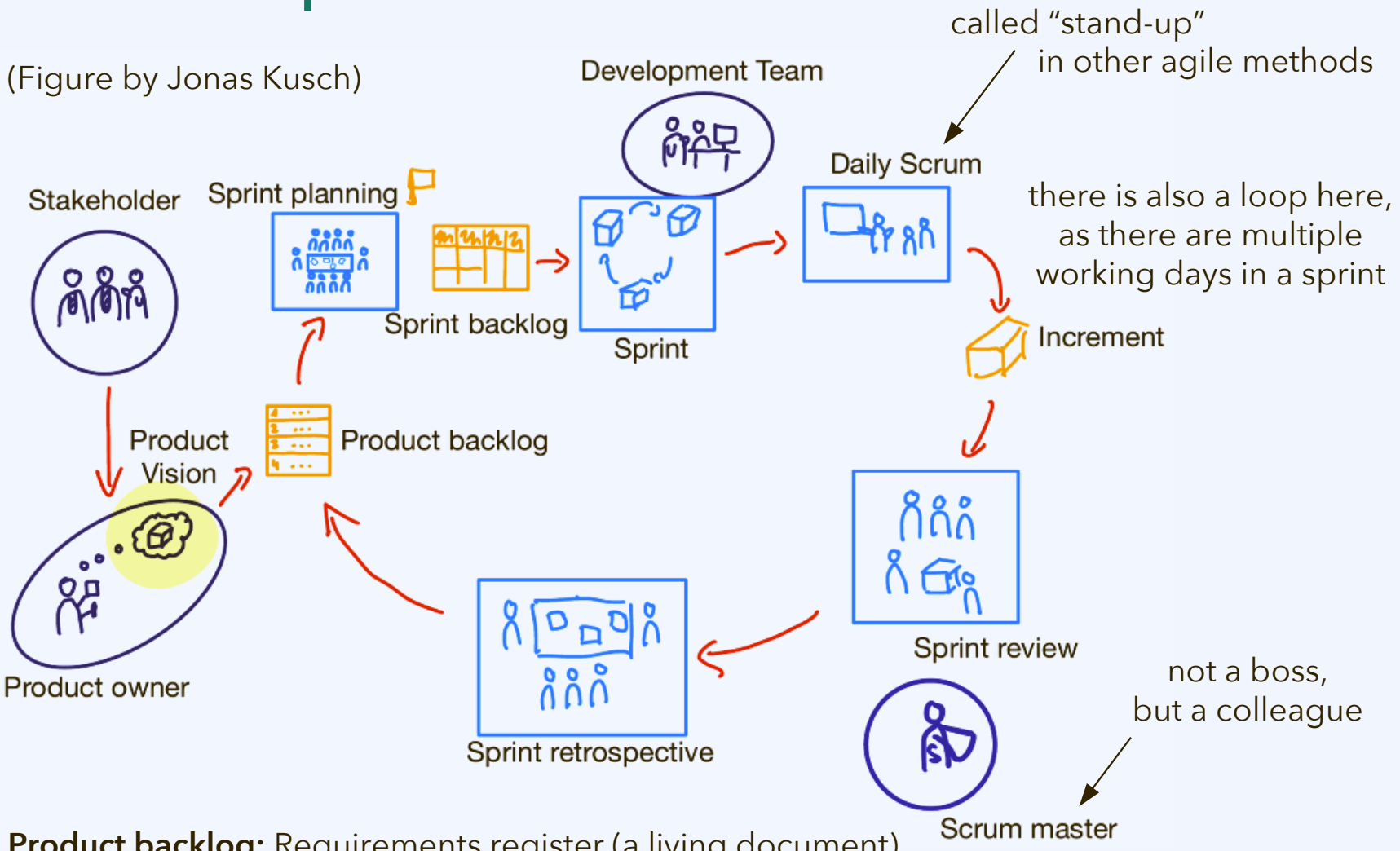
- Staff are subject to traditional hierarchies and reporting.
- They must pretend to be agile and implement agile-looking methods.
- This must also be documented, of course, and reported to the bosses.
- As a result, the agile methods don't simplify things, quite the opposite.

Example from a previous job:

- Idea: Do "agile stand-ups" – daily meetings of team with scrum master, where people determine flexibly what they will work on that very day.
- Real life: No scrum master, always the same boss. No flexible action, since people are working on different projects. Instead, it becomes reporting to the boss: "What did you accomplish in the last 24 hours?"

The scrum process

(Figure by Jonas Kusch)

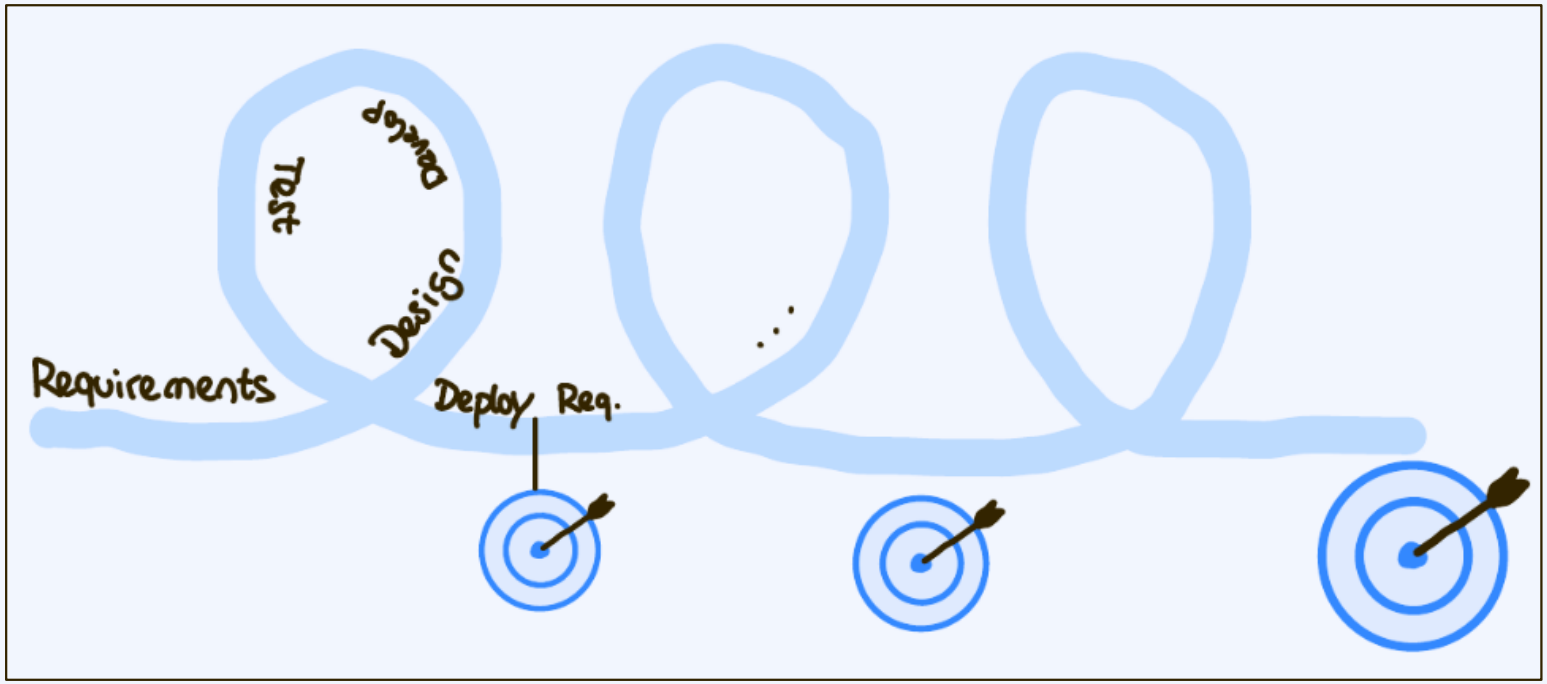


Product backlog: Requirements register (a living document).

Sprint backlog: Requirements that the team wants to cover within one specific sprint.

The scrum process

(Figure by Jonas Kusch)



In effect, similar to multiple waterfall projects, with a moving (growing) target.

The main advantages are that you have “something working” early on, but also that the “product owner” (*e.g.* customer) can rethink what is really needed.

Agile methodology in INF203

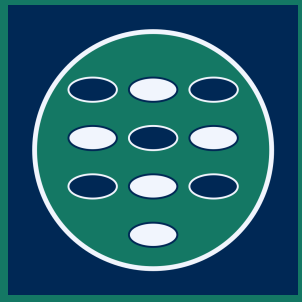
In what ways does our work in INF203 reflect agile principles?

- If you follow the sequence of **worksheets**, you can do four **sprints**. Each leads to a version of the code with a greater set of features.
- Every year, the problem in INF203 is a **new** one. Much is up to **discussion**, it is a bit chaotic, not everything is directly clear and well-defined.
- It is up to you if you use a development method inspired by **scrum**, extreme programming (*e.g.* **pair programming**), or other agile paradigms. You don't have to, but it is encouraged.
- There is one concrete agile method that all will work with: Agile requirements analysis based on **epics** and **user stories**.

What do you think: In what ways does INF203 *not* reflect agile principles?

Noregs miljø- og biovitskaplege universitet

Institutt for datavitskap



Digitalisering på Ås

INF203

June advanced programming project

1 Introduction

1.1 Course structure
1.2 June problem intro
1.3 Group management
1.4 Agile methods
1.5 Requirements analysis
1.6 Good documentation