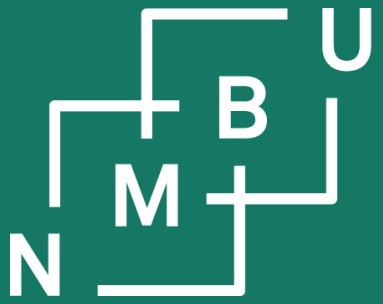
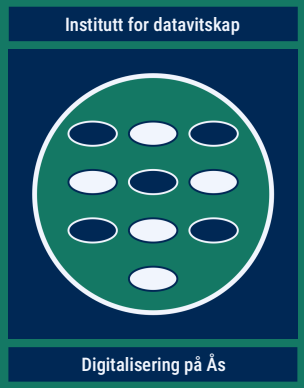


Noregs miljø- og biovitskaplege universitet



INF203

June advanced programming project

4 Development for production

The “colloquium” next week

- It is like an oral exam for the group, contributing 30% of the grade.
- It is technically not an oral exam, since INF203 has portfolio evaluation (gjennomgående vurdering), so it is a part of the evaluated portfolio.
- Be very sure that you know when your colloquium has been scheduled.
- All group members must attend, it is possible to join remotely via zoom or teams if you have a good reason. (Please explain and arrange in advance.) The external examiner will also join through the remote link.
- In the beginning (max. 10 minutes), you have the opportunity to start by presenting something. It is completely up to you how you use this opportunity, and this is not an obligation, we can also just start talking. It is not a problem it you present less than ten minutes.

Important info about the deadline

The advanced programming project INF203 is in part about "**advanced programming**," but it is mostly about doing a "**programming project**," carrying out a project as a group.

In view of this, **keeping the deadline** is important, as this is also needed in "real life." But we also know that in reality, many deadlines are negotiable.

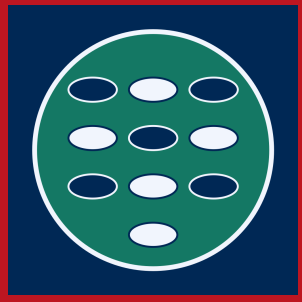
In our case, there are two reasons why the final deadline (21.6. 17.30) is non-negotiable and anything submitted later cannot be considered.

1) The submissions need to be prepared for access by the external examiner, who cannot just load them from Canvas. This must be before we have our colloquia.

2) After the deadline, groups are not obliged to keep their code confidential. So if they want, they can *e.g.* open their repositories to the public or upload code to TestPyPI. This is no problem for our teaching in the future. INF203 has a new problem each year.

Noregs miljø- og biovitskaplege universitet

Institutt for datavitskap



Digitalisering på Ås

4 Production

4.5 Reference data

4.6 Scientific computing

INF203

18th June 2025

Vrabec *et al.* (2006)

Vapour–liquid coexistence of the truncated and shifted Lennard–Jones fluid

Table 2. Saturated densities and interface properties from planar interface VLE simulations. The number in parentheses denotes the uncertainty in the last digit.

T^*	l^*	N	ρ_l^*	ρ_v^*	D^*	γ^*
0.625	11.85	2759	0.8244	0.0024	1.5658	0.727 (9)
0.650	11.91	2769	0.8120	0.0036	1.6616	0.675 (9)
0.675	11.97	2769	0.7994	0.0051	1.7494	0.633 (9)
0.700	12.04	2776	0.7863	0.0072	1.8621	0.581 (8)
0.725	12.11	2782	0.7729	0.0095	1.9844	0.549 (9)
0.750	12.18	2797	0.7599	0.0122	2.1065	0.493 (8)
0.775	12.26	2852	0.7452	0.0153	2.2517	0.455 (8)
0.800	12.34	2872	0.7307	0.0203	2.4348	0.403 (7)
0.825	12.43	2888	0.7156	0.0244	2.5695	0.369 (7)
0.850	12.52	2897	0.6993	0.0302	2.7943	0.322 (7)
0.875	12.63	2961	0.6821	0.0368	3.0027	0.275 (6)
0.900	12.75	2958	0.6632	0.0443	3.3181	0.239 (6)
0.925	12.87	3065	0.6437	0.0508	3.6935	0.191 (6)
0.950	13.01	3141	0.6222	0.0635	4.0550	0.156 (6)
0.975	13.18	3315	0.5987	0.0793	4.6938	0.122 (5)
1.000	13.39	3538	0.5672	0.0891	5.4502	0.081 (5)
1.025	13.66	3597	0.5265	0.1064	7.1065	0.057 (4)
1.050	14.05	4027	0.4827	0.1233	8.1196	0.030 (4)

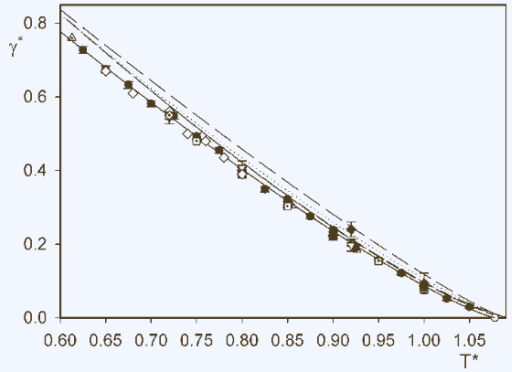


Figure 9. Surface tension vs. temperature for the truncated and shifted Lennard–Jones fluid: (●) results from planar interface VLE simulations (this work), (○) critical point, (——) correlation (equation (15)). Simulation results of other authors: (⊡) Haye and Bruin [10], (◇) Mareschal *et al.* [5], (◈) Holcomb *et al.* [15], (◡) Chen [14], (▽) Trokhymchuk and Alejandre (Monte Carlo) [12], (◯) Trokhymchuk and Alejandre (molecular dynamics) [12], (◆) Nijmeijer *et al.* [13], (△) Dunikov *et al.* [11]. Experimental real substance data reduced with the potential parameters from table 3: (– · –) argon [40], (· · · · ·) krypton [41], (- - - -) xenon [41], (– – –) methane [42].

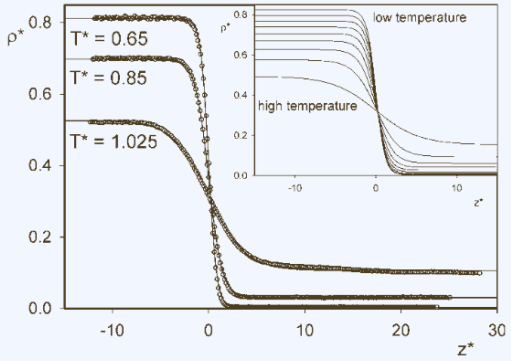


Figure 8. Large plot: density profiles of the planar interface for three different temperatures: (○) simulation results, (——) correlation (equations (7), (8), and (12)). Small plot: same correlation over the whole simulated temperature range $T^* = 0.625$ to 1.05.

Simulation data were correlated by

$$\gamma^* = 2.08(1 - T^*/T_c^*)^{1.21}, \qquad (15)$$

where the empirically found exponent 1.21 is close to the value of the 'critical index' 1.26 from fluctuation theory of critical phenomena [32].

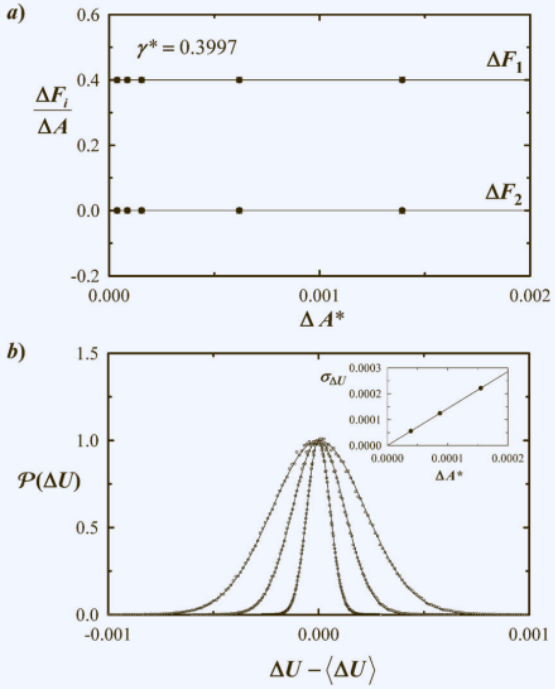
J. Vrabec *et al.*, *Mol. Phys.* **104**(9): 1509–1527, doi:10.1080/00268970600556774, **2006**.

Sampayo *et al.* (2010)

THE JOURNAL OF CHEMICAL PHYSICS **132**, 141101 (2010)

Communications: Evidence for the role of fluctuations in the thermodynamics of nanoscale drops and the implications in computations of the surface tension

José G. Sampayo,[1] Alexandr Malijevský,[1,2] Erich A. Müller,[1] Enrique de Miguel,[3] and George Jackson[1,a)]

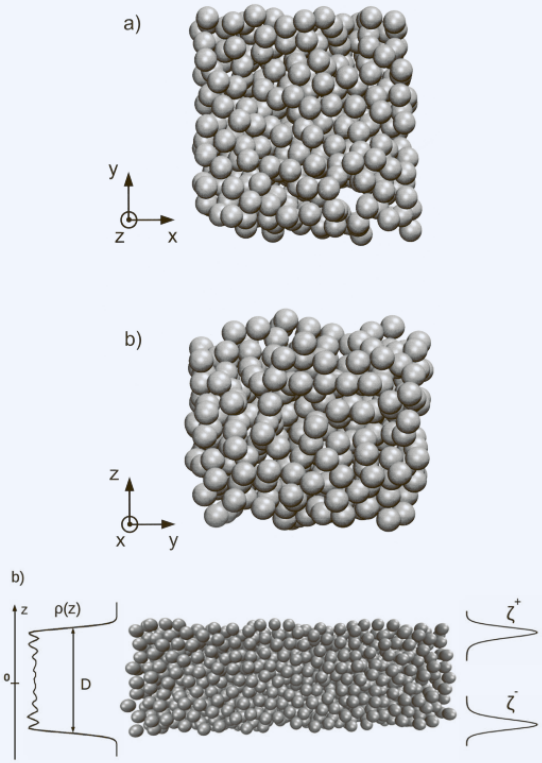


Test-area deformations are used to analyze vapor-liquid interfaces of Lennard-Jones particles by molecular dynamics simulation. For planar vapor-liquid interfaces the change in free energy is captured by the average of the corresponding change in energy, the leading-order contribution. This is consistent with the commonly used mechanical (pressure-tensor) route for the surface tension. By contrast for liquid drops, one finds a large second-order contribution associated with fluctuations in energy. Both the first- and second-order terms make comparable contributions, invalidating the mechanical relation for the surface tension of small drops. The latter is seen to increase above the planar value for drop radii of ~8 particle diameters, followed by an apparent weak maximum and slow decay to the planar limit, consistent with a small negative Tolman length.

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 $L_z'=L_z/(1+\xi)$. (a) The contributions $\Delta F_1/\Delta A$ and $\Delta F_2/\Delta A$ to the change in free energy per unit area (in units of ϵ/σ^2) ($\Delta A^*<0$, +; $\Delta A^*>0$, ×; and average, ●). The interfacial tension $\gamma^*=\gamma\sigma^2/\epsilon$ is obtained by extrapolation to $\Delta A^*=\Delta A/\sigma^2=0$. (b) 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.

J. S. Sampayo *et al.*, *J. Chem. Phys.* **132**: 141101, doi:10.1063/1.3376612, **2010**.

Filippini *et al.* (2014)

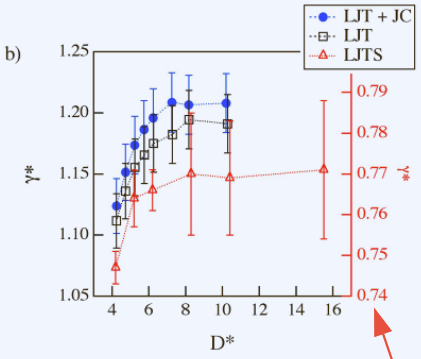


THE JOURNAL OF CHEMICAL PHYSICS **141**, 081103 (2014)

Communication: Slab thickness dependence of the surface tension: Toward a criterion of liquid sheets stability

G. Filippini,[1] E. Bourasseau,[1,a)] A. Ghoufi,[2] F. Goujon,[3] and P. Malfreyt[3]
[1]*CEA/DAM/DIF, F-91297 Arpajon Cedex, France*
[2]*IPR, UMR CNRS 6251, 263 avenue du Général Leclerc, 35042 Rennes, France*
[3]*ICCF, UMR CNRS 6296, BP 1048, F-63000 Clermont-Ferrand, France*

(Received 14 May 2014; accepted 20 August 2014; published online 29 August 2014)



reduced surface tensions calculated using different modified Lennard-Jones potentials

tials. In Fig. 4, we have plotted the γ/γ_∞ for all the potentials used, where γ_∞ is the surface tension for the largest sheet.

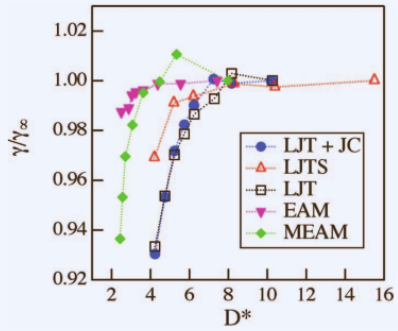


FIG. 4. Ratio of the surface tension to the surface tension γ_∞ calculated at the largest D^* for various types of potentials.

These numbers look strange, and they don't make it very clear what temperature they are simulating their LJTS system at.

G. Filippini *et al.*, *J. Chem. Phys.* **141**: 081103, doi:10.1063/1.4894399, **2014**.

Stephan *et al.* (2018)

RESULTS AND DISCUSSION

Surface Tension. Figure 1 shows the results for the surface tension γ^* of the LJTS fluid as a function of the temperature

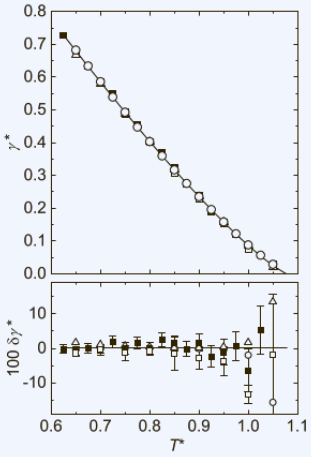


Figure 1. Top: Surface tension of the LJTS fluid as a function of the temperature. Data obtained from molecular simulations (filled squares: ref 28; empty squares: this work), DGT (circles), and DFT (triangles). The solid line is an empirical correlation for the surface tension proposed in ref 28, cf. eq 13. Bottom: Relative deviation for the surface tension from the empirical correlation: values by DGT (circles), DFT (triangles), MD from this work (empty squares), and Vrabec et al.[28] (full squares) as a function of temperature.

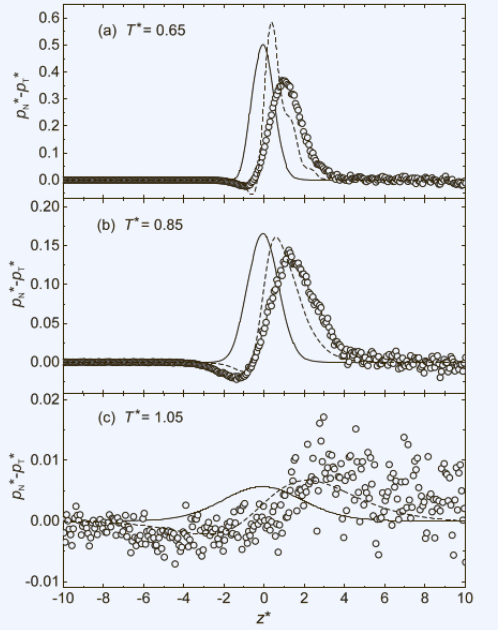


Figure 2. Difference between normal and tangential pressure at interface of the LJTS fluid at (a) $T^* = 0.65$; (b) $T^* = 0.85$; and $T^* = 1.05$: molecular simulations (symbols), DGT (solid line), a DFT (dashed line).

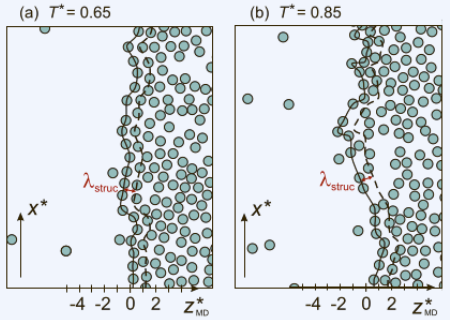


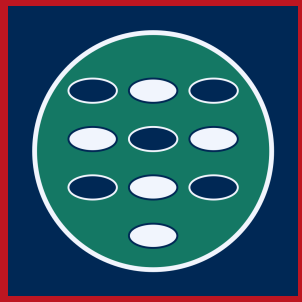
Figure 7. Snapshots of the vapor–liquid interface from molecular simulations of the LJTS fluid at (a) $T^* = 0.65$ and (b) $T^* = 0.85$. The slice that is shown has a width of $\Delta y^* = 1$. The solid black line schematically indicates the intrinsic surface from the current molecular configuration. The black dotted line indicates the equidistant surface for the layering structure at the distance λ^*_{struc} from the intrinsic surface.

Table 3. Surface Tension Data from MD, DGT, and DFT

T^*	γ^*_{MD}	γ^*_{DGT}	γ^*_{DFT}
0.65	0.670(7)	0.683	0.669
0.7	0.584(7)	0.586	0.579
0.75	0.486(11)	0.493	0.489
0.8	0.400(4)	0.403	0.402
0.85	0.312(15)	0.317	0.317
0.9	0.229(7)	0.235	0.235
0.95	0.152(7)	0.158	0.157
1	0.075(2)	0.088	0.085
1.05	0.025(4)	0.029	0.022

S. Stephan *et al.*, *J. Phys. Chem. C* **122**(43): 24705–24715, doi:10.1021/acs.jpcc.8b06332, **2018**.

Noregs miljø- og biovitskaplege universitet



4 Production

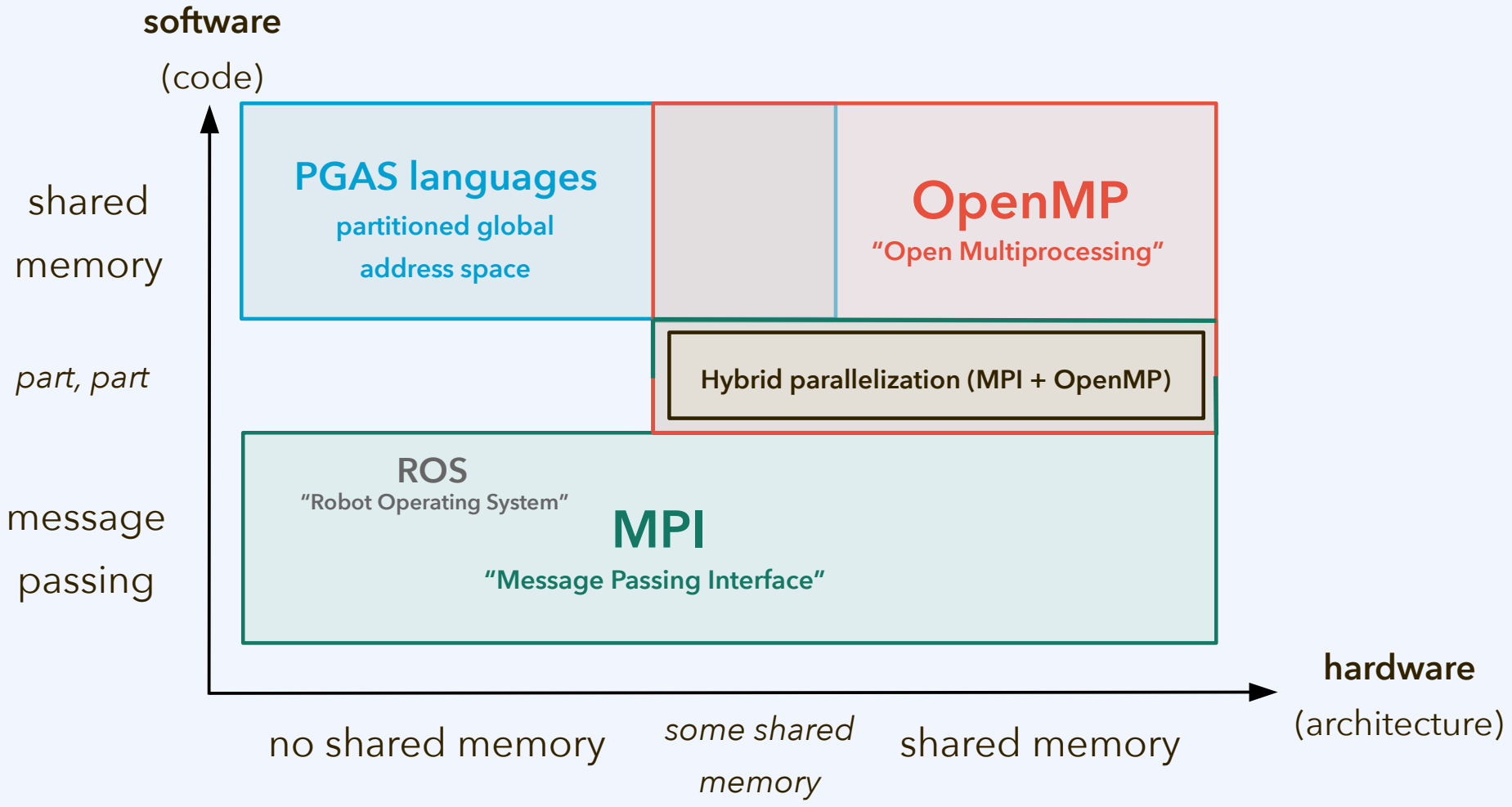
4.5 Reference data

4.6 Scientific computing

INF203

18th June 2025

Parallel programming paradigms

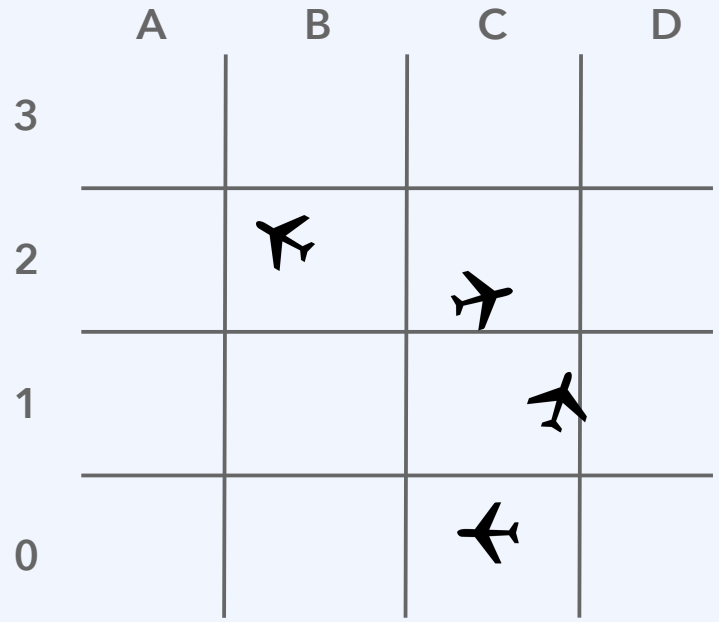


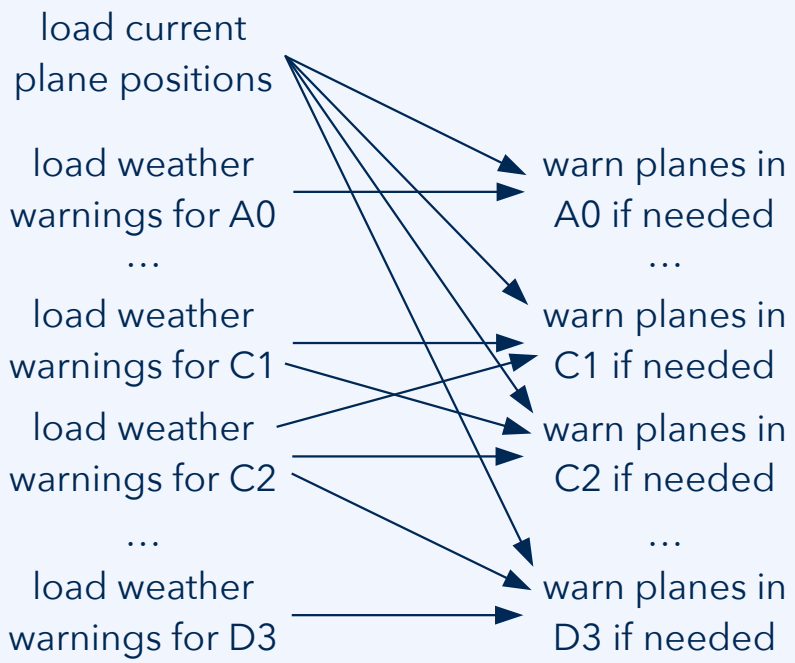
Spatial concurrency in the data

Domain decomposition is characterized by two features:

First, parallelization is based on the concurrency inherent in (some) data.

Second, these data are seen as constituting a space, or as located in a space.



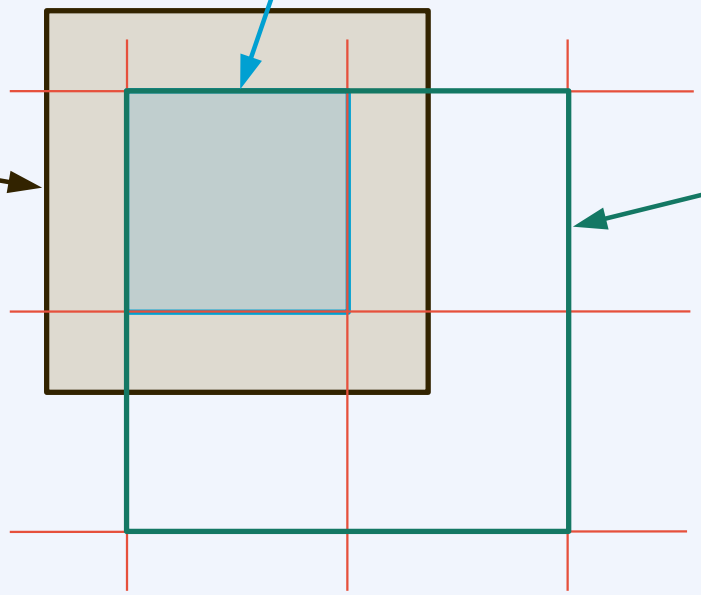


Example: Three-dimensional box

one of the **local boxes** into which the system is divided for parallelization

halo region of the loxal box: the process is not responsible for this information, but needs to know it

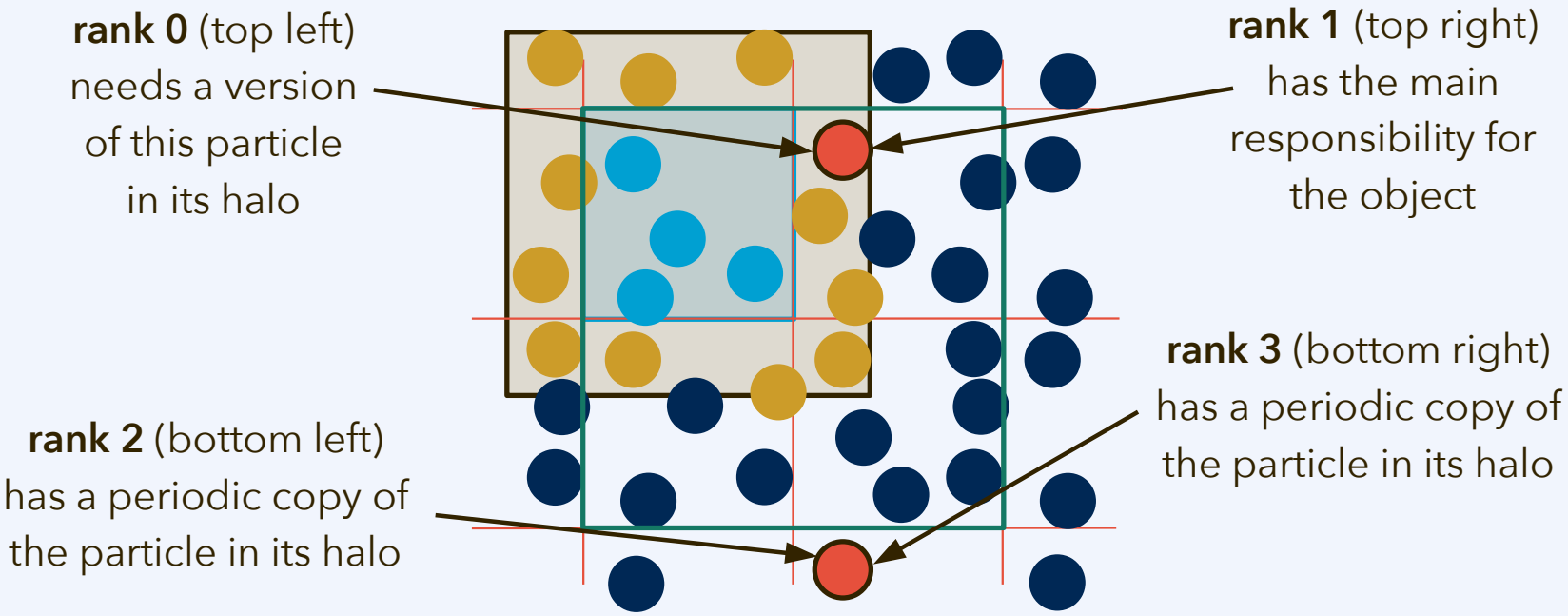
subdomain from MPI rank:



global 3D system containing all the "original" versions of the particles

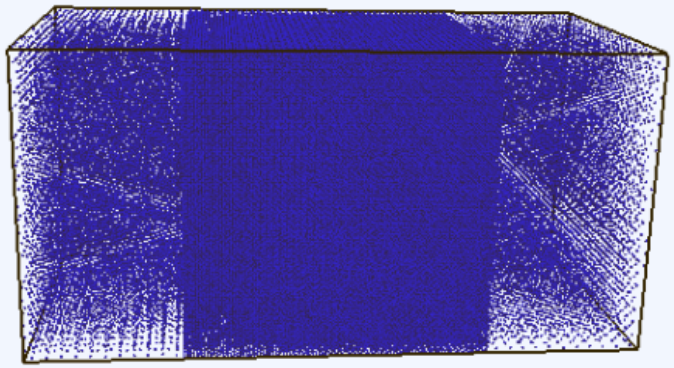
Example: Three-dimensional box

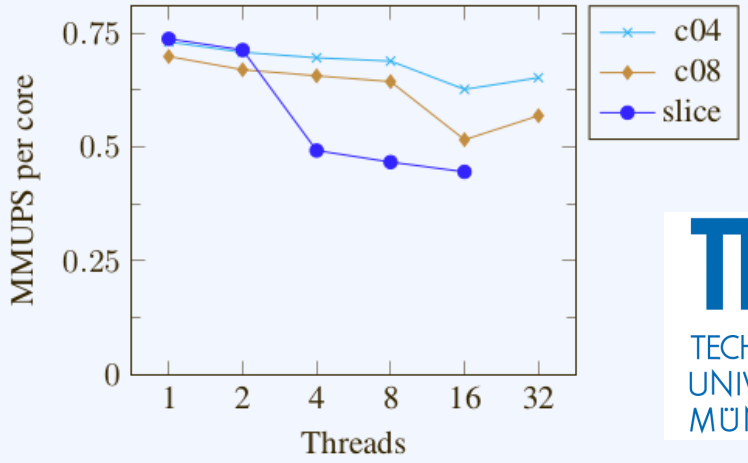
For a single particle read in from the input file, **multiple copies** can now exist in several ranks.



If a molecule is *updated or moved*, adjacent ranks may need to be informed.

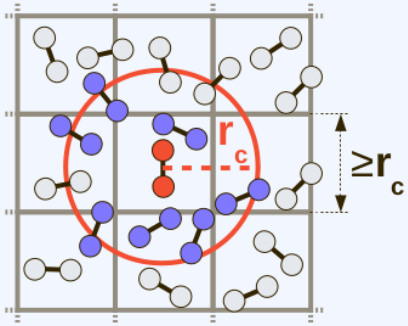
Molecular dynamics world record

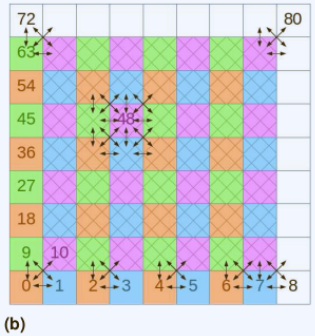




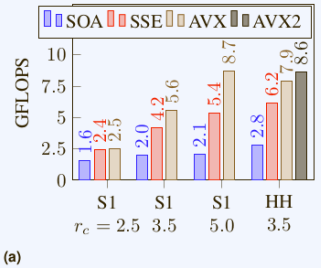
(a) (b)

Figure 5. (a) Visualization of load imbalance scenario, (b) strong scaling on one node of S1 for the scenario visualized in (a). One thread per core is used, except for the case with 32 threads, which corresponds to hyperthreading (two threads per core).





(b)



(a)

Figure 3. (a) GFLOPS performance for varying r_c (in σ)

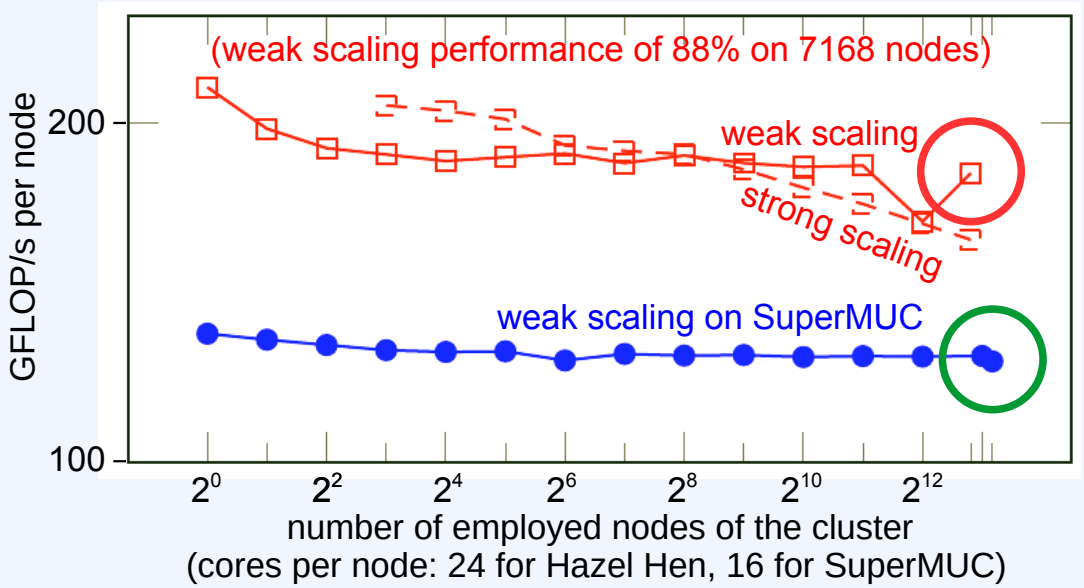
Computational Molecular Engineering

[1]N. Tchipev *et al.*, *Int. J. HPC Appl.* **33**(5): 838–854, doi:10.1177/1094342018819741, **2019**.

Molecular dynamics world record

Hazel Hen (Stuttgart):
Haswell architecture

http://www.ls1-mardyn.de/ (large systems 1: molecular dynamics)



N = 21 000 000 000 000
2019 molecular dynamics world record[1]

Hazel Hen
weak scaling

TECHNISCHE UNIVERSITÄT MÜNCHEN

UHH

Computational Molecular Engineering

[1]N. Tchipev *et al.*, *Int. J. HPC Appl.* **33**(5): 838–854, doi:10.1177/1094342018819741, **2019**.

Load balancing

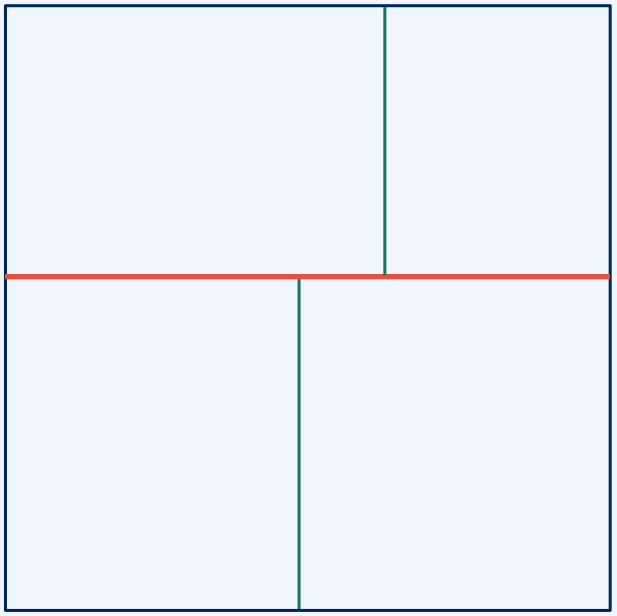
Requirements modelling can be used to predict how the way in which the domain is decomposed, for a given input case/scenario, influences the load of each of the parallel processes.

Usually, no quantitatively accurate requirements model exists. Even then, a rough approximation can be used as guidance for distributing the load.

Example scheme: **Recursive bisection** (with "*k*-dimensional tree," $k = 3$).

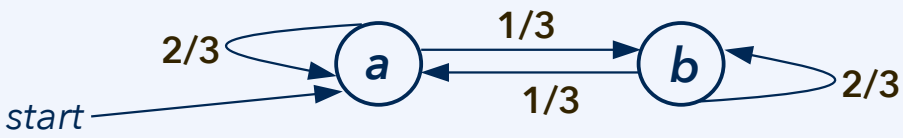
From the top (whole domain) down to the bottom (single process), split the volume recursively into parts such that processes will receive a similar load.

Alternate between spatial dimensions.



Concurrency in Markov chains, random walks, *etc.*

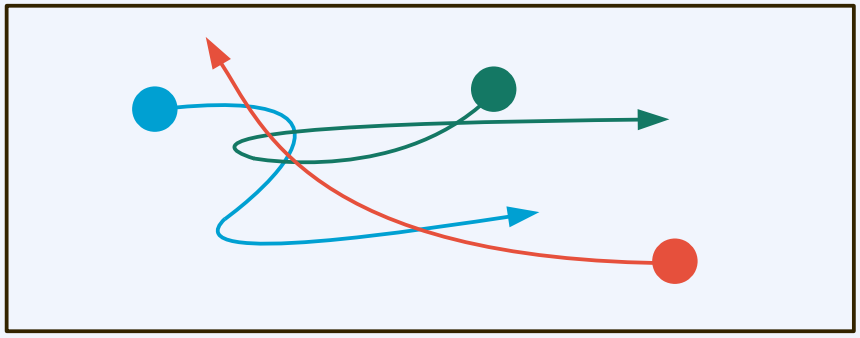
A Markov chain is a sequence of states in a probabilistic discrete event system.



abbabbbaaba …
ababbaaabba …

The sequence of configurations in a Metropolis Monte Carlo simulation is such a Markov chain. So are many variants of it, or other common solutions to problems that require the **stochastic exploration** or sampling of a **large space**.

The *concurrency* here is due to that *multiple Markov chains are independent*.



Processes/threads can explore the state space separate from each other. They work with independent configurations. It is not necessary to implement a domain decomposition.

FYS253: Thermal physics (autumn)

Norwegian University of Life Sciences

FYS253

Thermal Physics

Credits (ECTS):
5

Course responsible:
Mathijs Adriaan Janssen

Campus / Online:
Taught campus Ås

Teaching language:
Engelsk

Course frequency:
Annually

Nominal workload:
For 13 weeks: Lectures: 1 × 2 hours per week. Exercises: 1 × 2 hours per week. Individual work and group discussions: 5 hours per week for 15 weeks. Total of 125 hours.

Teaching and exam period:
This course starts in autumn parallel. This course has teaching/evaluation in autumn parallel.

About this course

Teaching structure: One 2-hour lecture per week. One 2-hour problem solving class per week.

Content: Fundamental thermodynamics concepts and their application in different scenarios. First and second law, enthalpy, free energy, entropy, ideal gas mixtures, thermodynamics property relations,

TBM250: Finite element method (autumn)

Norwegian University of Life Sciences

TBM250

The Finite Element Method

Credits (ECTS):
10

Course responsible:
Eirik Valseth

Campus / Online:
Taught campus Ås

Teaching language:
Engelsk

Course frequency:
Annually

Nominal workload:
Lectures with self-tuition and homework, approx. 100 hours. Exercises and homework, approx. 150 hours.

Teaching and exam period:
This course has teaching/evaluation in the Autumn parallel

About this course

In this course an introduction to the classical finite element method for the solution of partial differential equations from engineering science, e.g., heat conduction, viscous flows, porous media flows, and elastic deformation. This will be achieved by considering two closely related aspects: i) The theoretical foundation of the finite element methods, and ii) use of existing finite element analysis software. Central topics are: the

INF205 and INF305 courses (spring)

5 ECTS each; INF205 in first half, INF305 in second half of the spring semester.

INF205

Resource-Efficient Programming

Credits (ECTS):
5

Course responsible:
Martin Thomas Horsch

Campus / Online:
Taught campus Ås

Teaching language:
Norsk

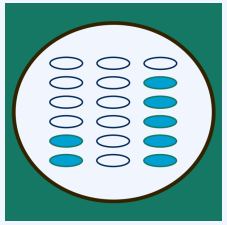
Course frequency:
Annually (spring semester, first half)

Nominal workload:
125h = 24h lectures + 12h computer lab + 89h self-study including work on programming tasks

Teaching and exam period:
The course is offered in the spring parallel. The course has teaching/assessment throughout the first half of the spring parallel.

About this course

This course introduces students with programming experience in high-level programming languages (e.g., Python) to programming in a compiled programming language with explicit memory management, with a focus on efficient use of computational resources (CPU time and memory). Specific topics are:



INF305

Scientific Computing

Credits (ECTS):
5

Course responsible:
Jonas Kusch

Campus / Online:
Taught campus Ås

Teaching language:
Engelsk

Course frequency:
Annually (spring semester, second half)

Nominal workload:
125h = 24h lectures + 12h exercise + 89h self-study including work on exercise tasks

Teaching and exam period:
The course is offered in the spring parallel. The course has teaching/assessment throughout the second half of the spring parallel.

About this course

This course introduces students to scientific computing, the collection of tools, techniques, and theories required to solve mathematical models of problems in Science and Engineering on the computer. A particular focus lies in theoretically understanding and efficiently implementing discussed algorithms to solve physical balance laws. The programming part covers essential and valuable programming paradigms required for

C++ vs. Python: Language features

C++	Python

"What do you know about language features that are different in C++ and Python?"

INF205: Structure of the course

1) C++ basics

- Getting started
- Procedural programming in C++ (not much different from Python)

2) Memory and objects

- Direct access to memory addresses, allocating and deallocating memory
- Object oriented programming in C++ (somewhat different from Python)

3) Data structures and libaries

- Containers (incl. lists, graphs), standard template library, other libraries
- Memory management for container data structures

(basic intro only – this is done in depth in the INF305 course)

4) Concurrency

- Concurrent process models and *paradigms of parallel programming*
- Using the ROS2 robot operating system from within C++

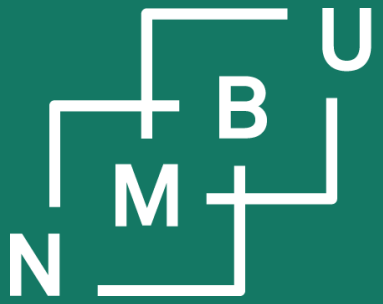
INF205: Learning outcomes

After completing the course you will be able to

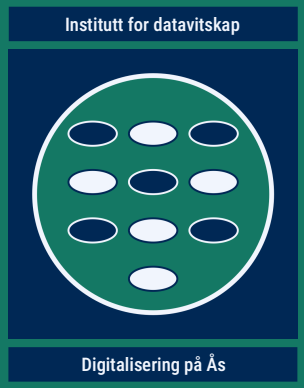
- implement solutions in modern C++;
- manage memory safely;
- make use of capabilities provided by the C++ Standard Library and third-party libraries;
- implement data types from “first principles;”
- assess programs and their use in terms of sustainability metrics;
- write code suitable for concurrent processes and embedded systems;
- create interfaces allowing your code to interact with other software.

We speak of “**modern C++**” because of the long history of C++, *e.g.*, retaining all of the C programming language. C++ is like several languages in one.

Focus: Develop solutions that work both reliably and efficiently.



Noregs miljø- og biovitskaplege universitet



INF203

June advanced programming project

4 Development for production