Density-Functional Theory for Practitioners - Tutorial 5

Orlando Silveira Júnior, Adolfo Otero Fumega and Ondřej Krejčí,
(Developed by Patrick Rinke and Milica Todorović)

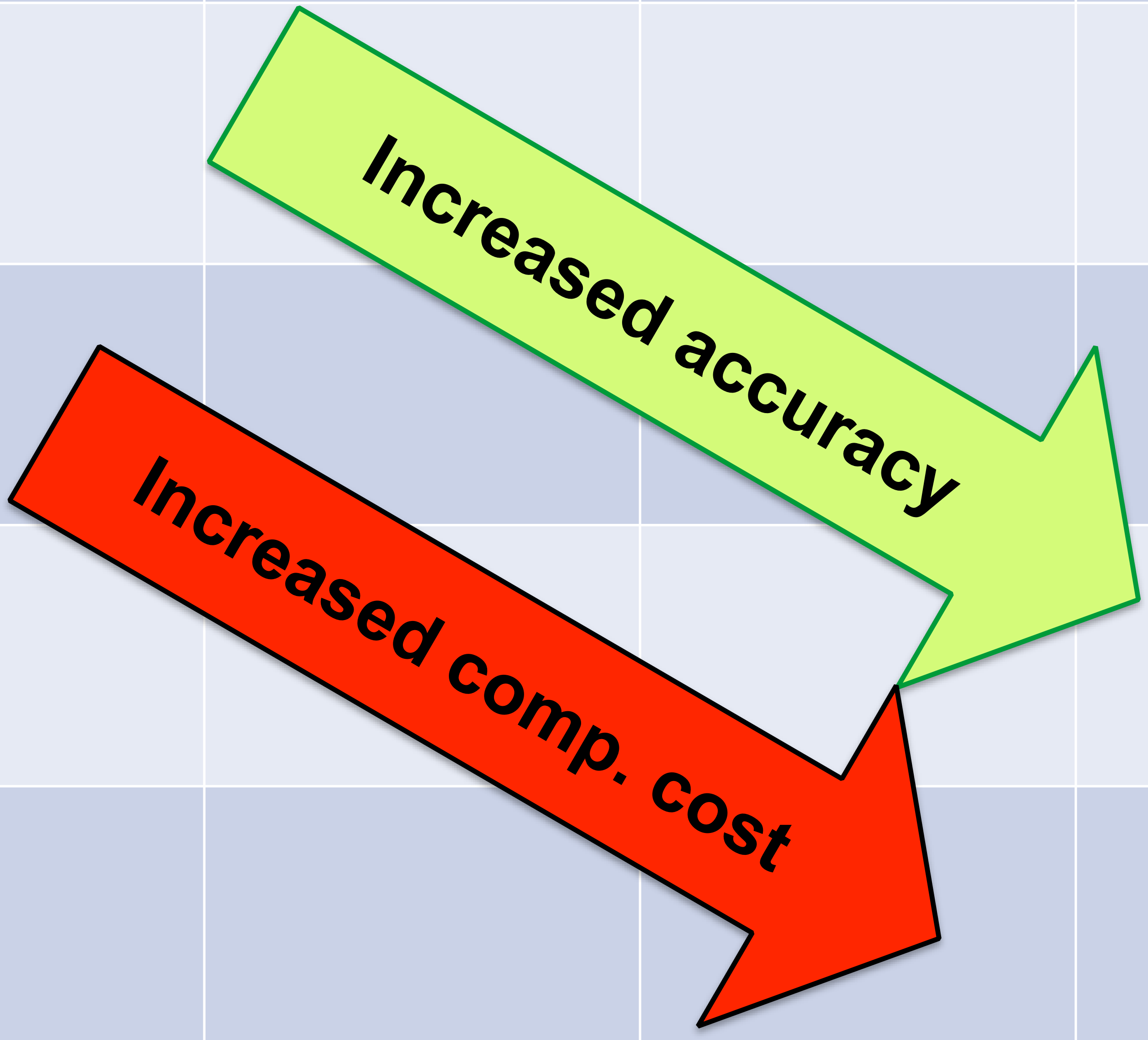
Aalto University
School of Science
Department of Applied Physics

Please read (and fill) the handouts in the meantime

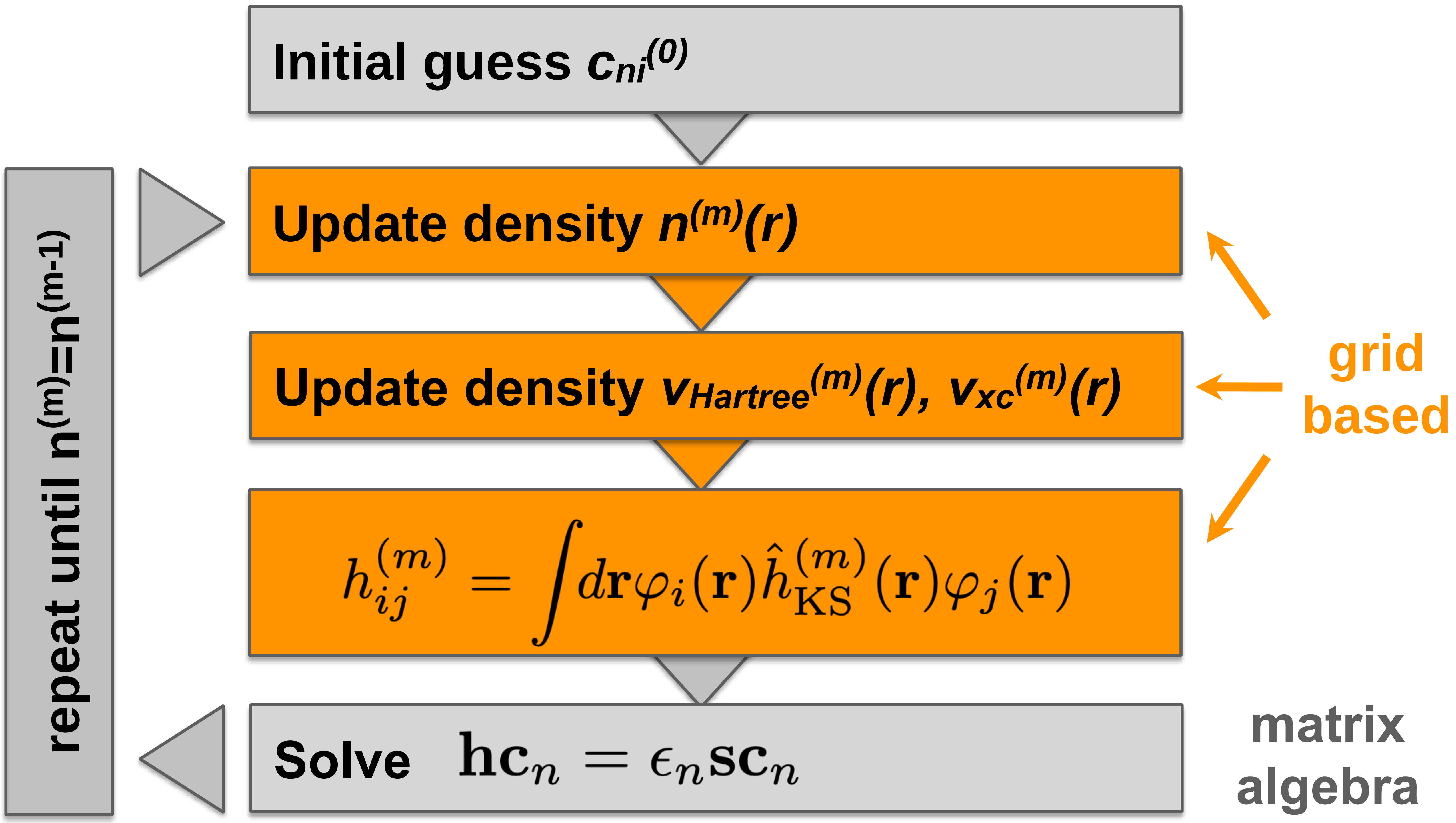
Aalto University
School of Science

FHI-aims basis set overview

\Grids **Basis**	light	Inter-mediate hybrid-XC	tight	really tight
Tier 1				
Tier 2				
Tier 3				
Tier 4				

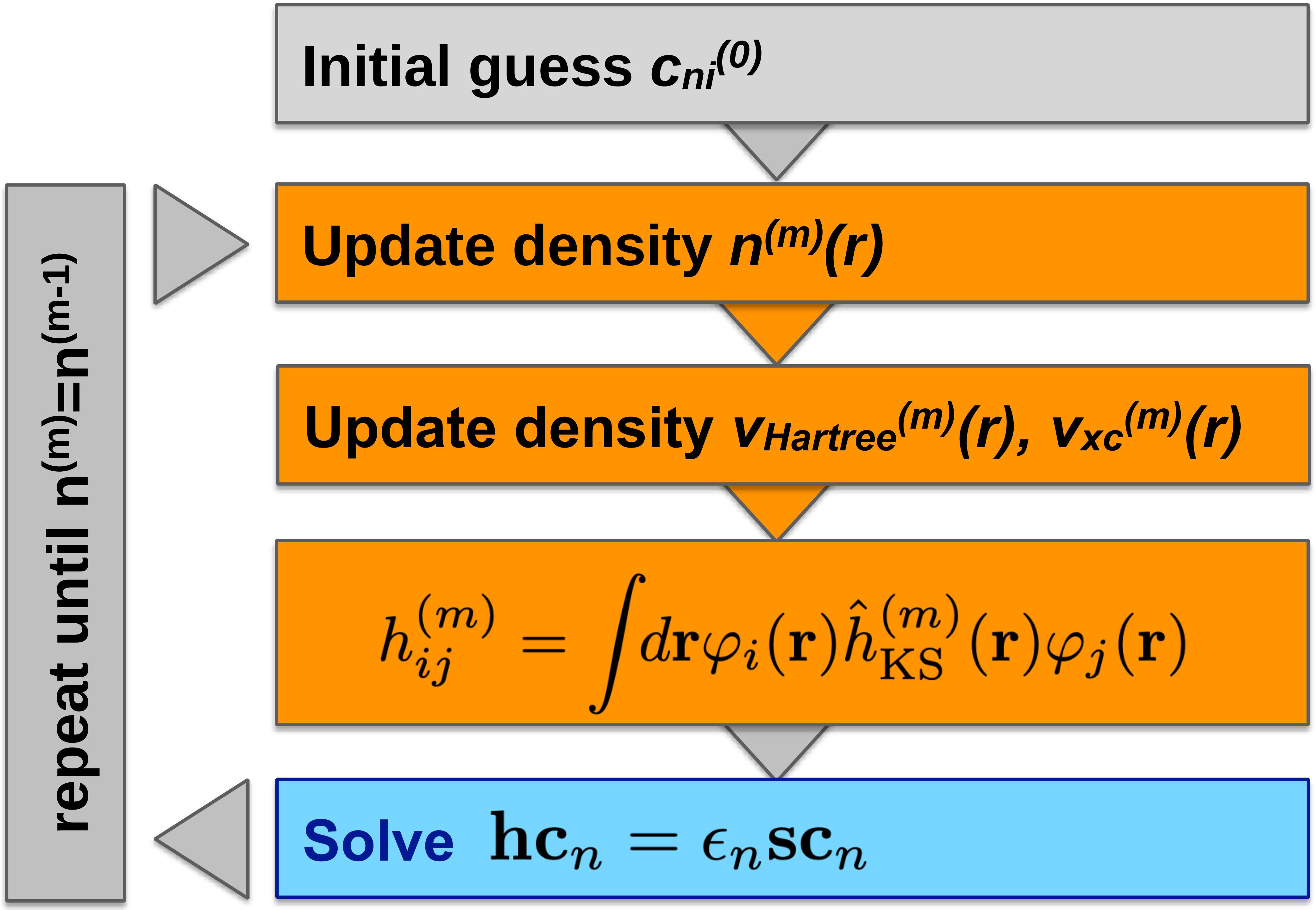


"Scaling" of DFT


Initial guess $c_{ni}^{(0)}$
Update density $n^{(m)}(r)$
Update density $v_{Hartree}^{(m)}(r)$, $v_{xc}^{(m)}(r)$
$h_{ij}^{(m)} = \int d\mathbf{r} \varphi_i(\mathbf{r}) \hat{h}_{\mathrm{KS}}^{(m)}(\mathbf{r}) \varphi_j(\mathbf{r})$
Solve $\mathbf{h}\mathbf{c}_n = \epsilon_n \mathbf{s}\mathbf{c}_n$
repeat until $n^{(m)}=n^{(m-1)}$
grid based
matrix algebra

Aalto University
School of Science

"Scaling" of DFT



Initial guess $c_{ni}^{(0)}$
Update density $n^{(m)}(r)$
Update density $v_{Hartree}^{(m)}(r)$, $v_{xc}^{(m)}(r)$
$h_{ij}^{(m)} = \int d\mathbf{r} \varphi_i(\mathbf{r}) \hat{h}_{\mathrm{KS}}^{(m)}(\mathbf{r}) \varphi_j(\mathbf{r})$
Solve $\mathbf{h}\mathbf{c}_n = \epsilon_n \mathbf{s}\mathbf{c}_n$
repeat until $n^{(m)}=n^{(m-1)}$

scaling critical step

Aalto University
School of Science

“Scaling” of DFT

$$\phi_n(\mathbf{r}) = \sum_i c_{ni}\varphi_i(\mathbf{r})$$

# basis function (N_b) increases with:

- # of atoms (N_a)
- Tier level

# of KS states increases with:

- # of electrons and thus # of atoms

Aalto University
School of Science

“Scaling” of DFT

$$\phi_n(\mathbf{r}) = \sum_i c_{ni}\varphi_i(\mathbf{r})$$

# basis function (N_b) increases with:

- # of atoms (N_a)
- Tier level

# of KS states increases with:

- # of electrons and thus # of atoms

matrix dimension $N_b \times N_b \Rightarrow N_a \times N_a$

$$h_{ij}c_{jn} = \epsilon_n s_{ij} c_{jn}$$

Aalto University
School of Science

Scaling of DFT

- Matrix <u>storage</u> is quadratic in N_b as well as in N_a.
- Matrix <u>inversion</u> is cubic in N_b as well as in N_a.

We call this the *formal* scaling of DFT.

matrix dimension $N_b \times N_b \Rightarrow N_a \times N_a$

$$\underbrace{h_{ij}}_{}c_{jn} = \epsilon_n s_{ij} c_{jn}$$

V. Blum et al. Computer Physics Communications 180, 2175 (2009)

Hand-outs on computational scalling

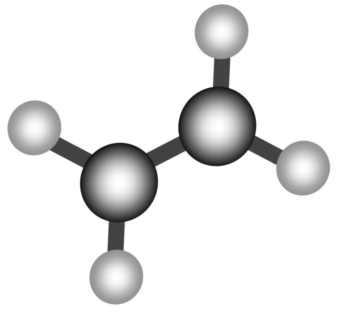
LUMI supercomputer at CSC

Scaling exercise 1 - finite polymer

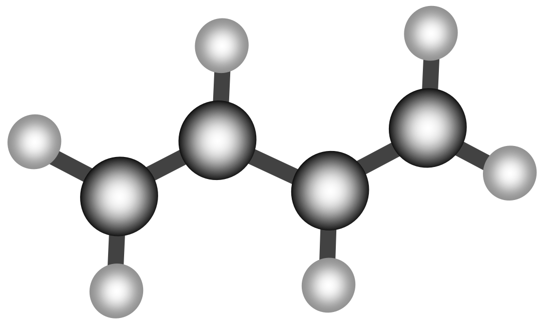
Instructions:

Fill in the table on the next page. Count the number of atoms in each system and write down by how much they increase compared to the first system (atomic factor). Then calculate the scaling factor and the estimated computer time, i.e. how much more expensive than system 1 is your new system.

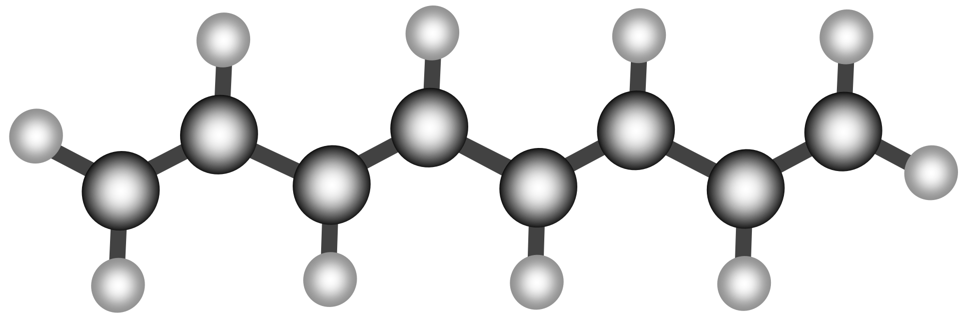
1:



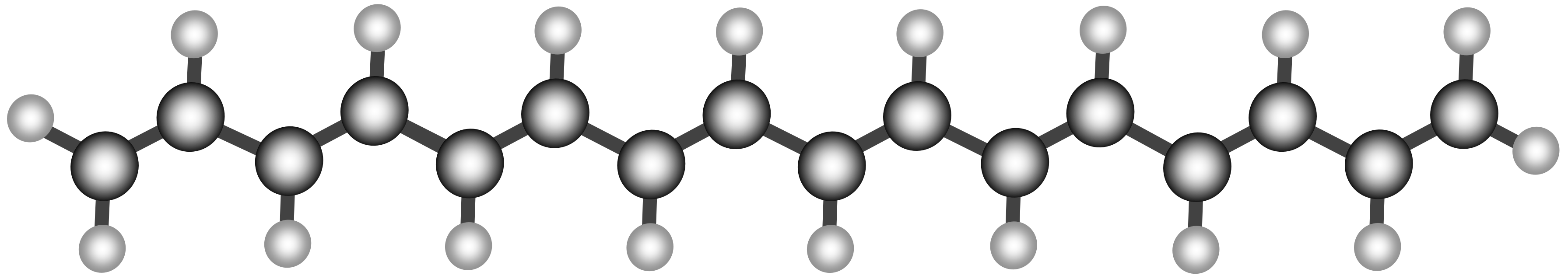
2:



3:

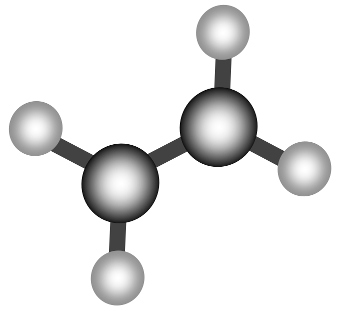


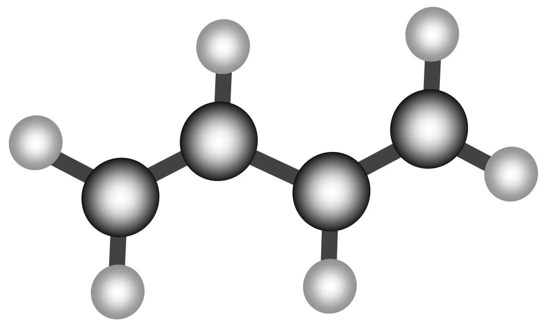
4:

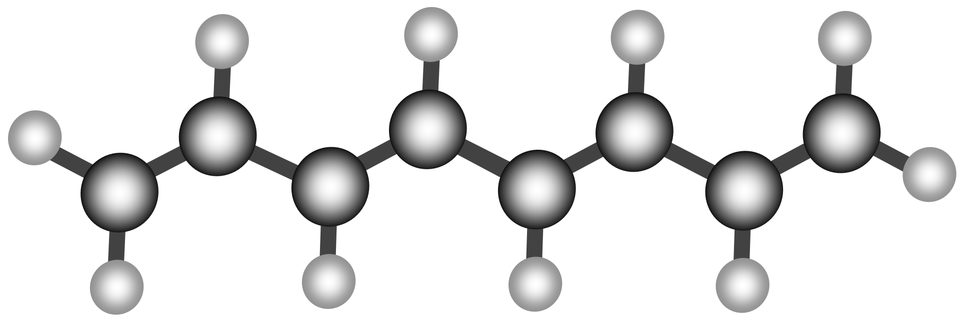


Scaling exercise 1 - finite polymer

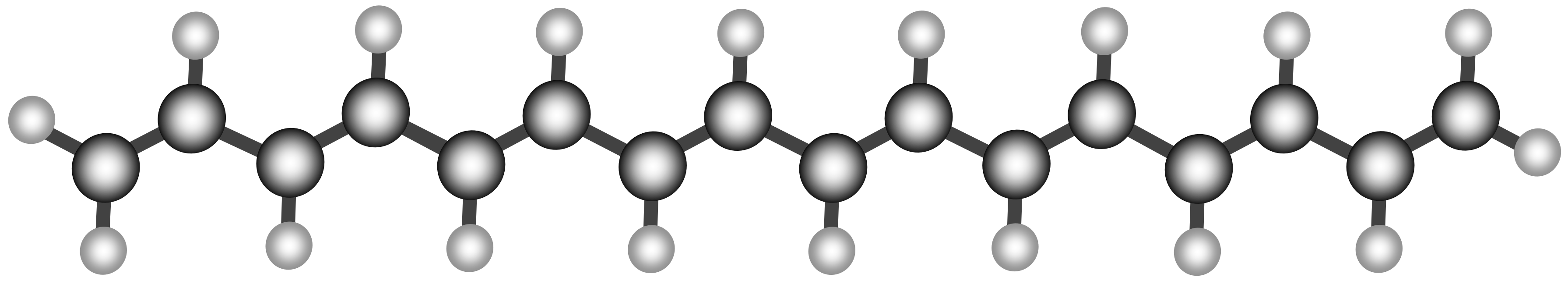
system	*# atoms*	*atomic factor*	*scaling factor*	*time*
1	*6*	*1*	*1*	*30s*
2	*10*	*~2*		
3				
4				

1: 

2: 

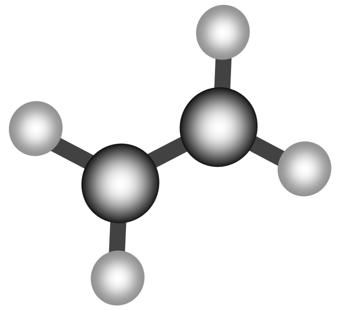
3: 

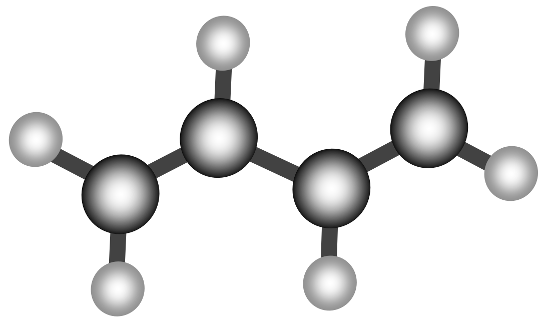
(You can round up the atomic factor, i.e. how many more atoms you have than in system 1, to make the scaling factor and time estimates easier.)

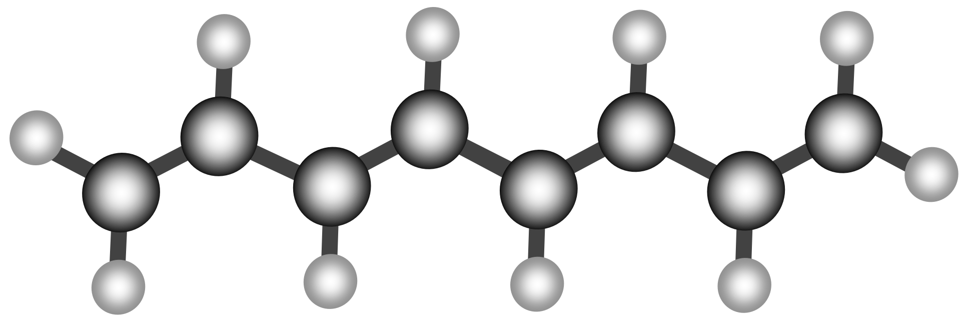
4: 

Scaling exercise 1 - finite polymer

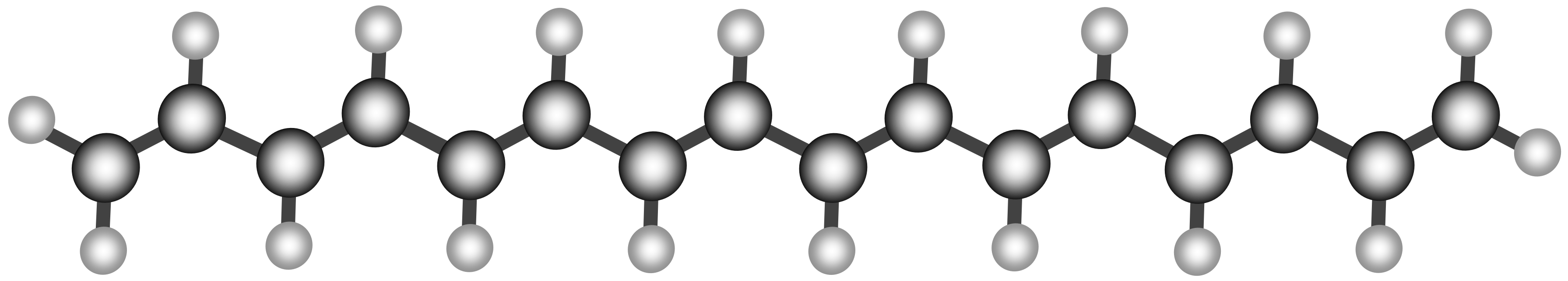
system	# atoms	atomic factor	scaling factor	time
1	6	1	1	30s
2	10	~2	~8	4min
3	18	3	27	13.5min
4	34	~6	~216	108min

1:


2:


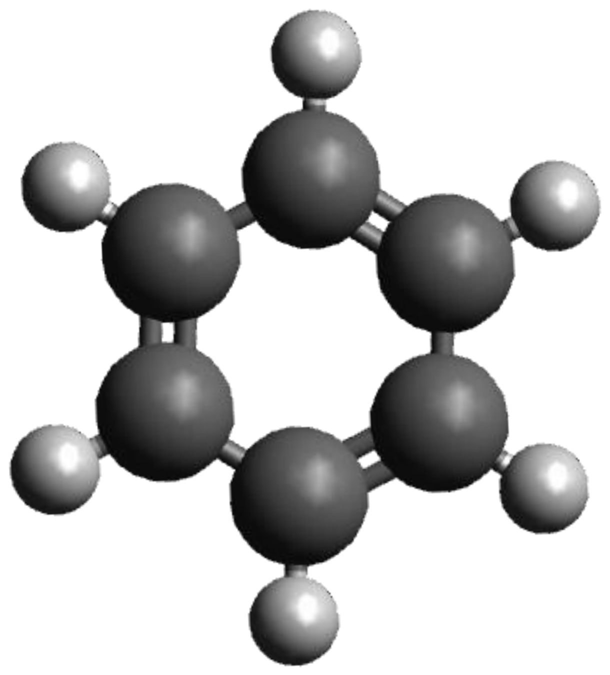
3:


(You can round up the atomic factor, i.e. how many more atoms you have than in system 1, to make the scaling factor and time estimates easier.)

4:


Scaling with basis functions

	H		C		O	
minimal	$1s$	**1**	[He]$+2s2p$	**5**	[He]$+2s2p$	**5**
Tier 1	H($2s$,2.1)		H($2p$,1.7)		H($2p$,1.8)	
	H($2p$,3.5)		H($3d$,6.0)		H($3d$,7.6)	
			H($2s$,4.9)		H($3s$,6.4)	
	4		**9**		**9**	
Tier 2	H($1s$,0.85)		H($4f$,9.8)		H($4f$,11.6)	
	H($2p$,3.7)		H($3p$,5.2)		H($3p$,6.2)	
	H($2s$,1.2)		H($3s$,4.3)		H($3d$,5.6)	
	H($3d$,7.0)		H($5g$,14.4)		H($5g$,17.6)	
			H($3d$,6.2)		H($1s$,0.75)	
	10		**25**		**25**	



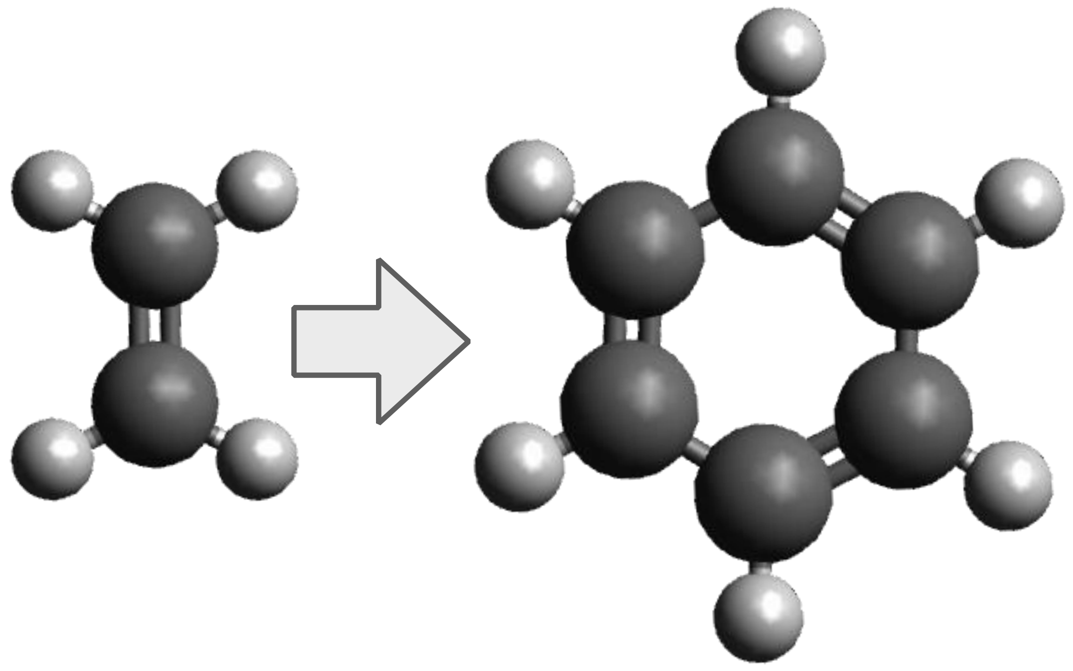
	N_b	N_b tot	cost
min	36	36	
Tier 1	78	114	x 32
Tier 2	210	310	x 20

Aalto University
School of Science

V. Blum et al. Computer Physics Communications 180, 2175 (2009)

Scaling with basis functions

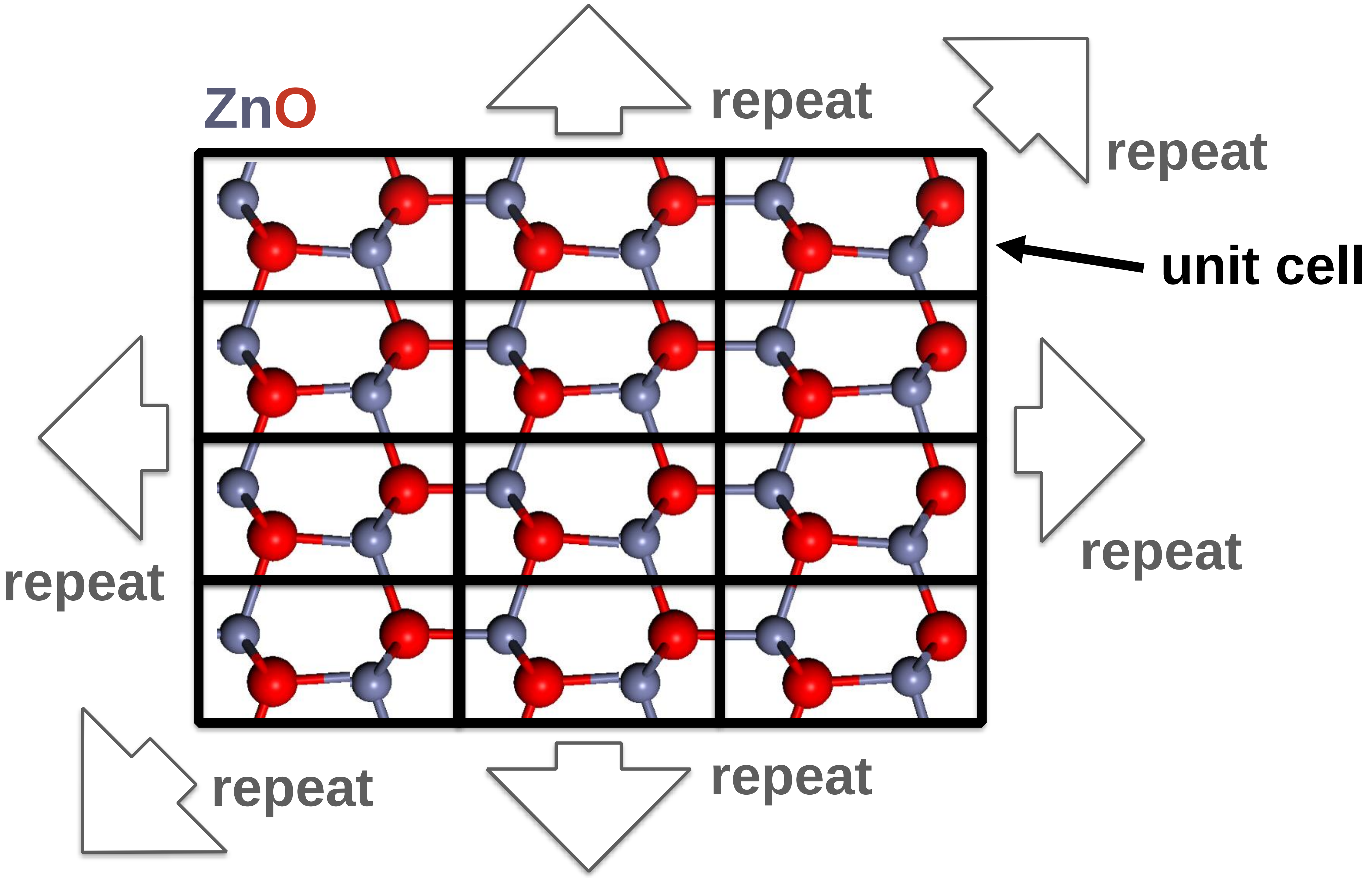
	H		C		O	
minimal	$1s$	1	[He]$+2s2p$	5	[He]$+2s2p$	5
Tier 1	H($2s$,2.1)		H($2p$,1.7)		H($2p$,1.8)	
	H($2p$,3.5)		H($3d$,6.0)		H($3d$,7.6)	
			H($2s$,4.9)		H($3s$,6.4)	
	4		9		9	
Tier 2	H($1s$,0.85)		H($4f$,9.8)		H($4f$,11.6)	
	H($2p$,3.7)		H($3p$,5.2)		H($3p$,6.2)	
	H($2s$,1.2)		H($3s$,4.3)		H($3d$,5.6)	
	H($3d$,7.0)		H($5g$,14.4)		H($5g$,17.6)	
			H($3d$,6.2)		H($1s$,0.75)	
	10		25		25	



Tier 1	N_b tot	cost
Ethylene	48	
Benzene	114	x 13

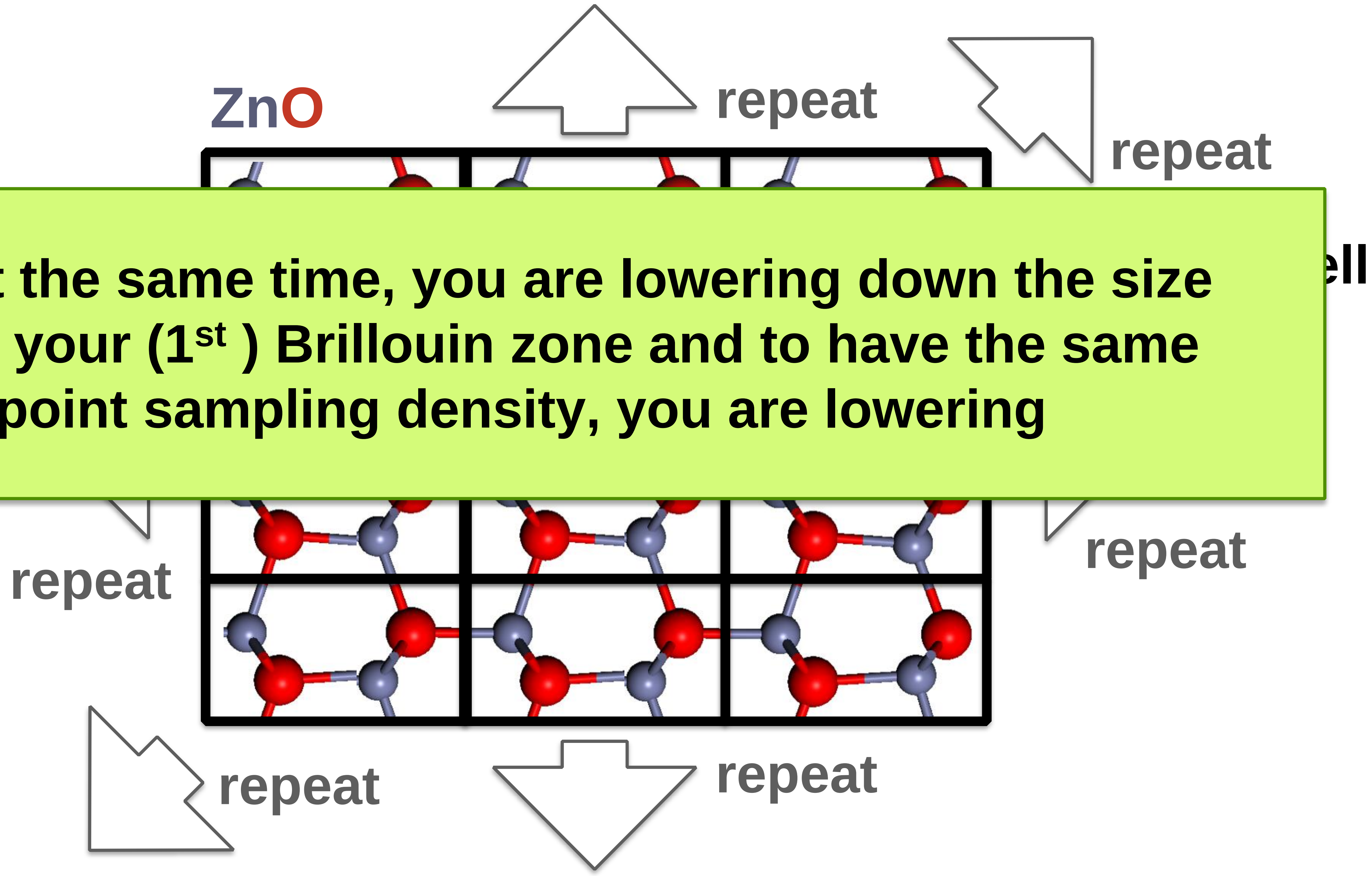
Aalto University
School of Science

Scaling in periodic boundary conditions



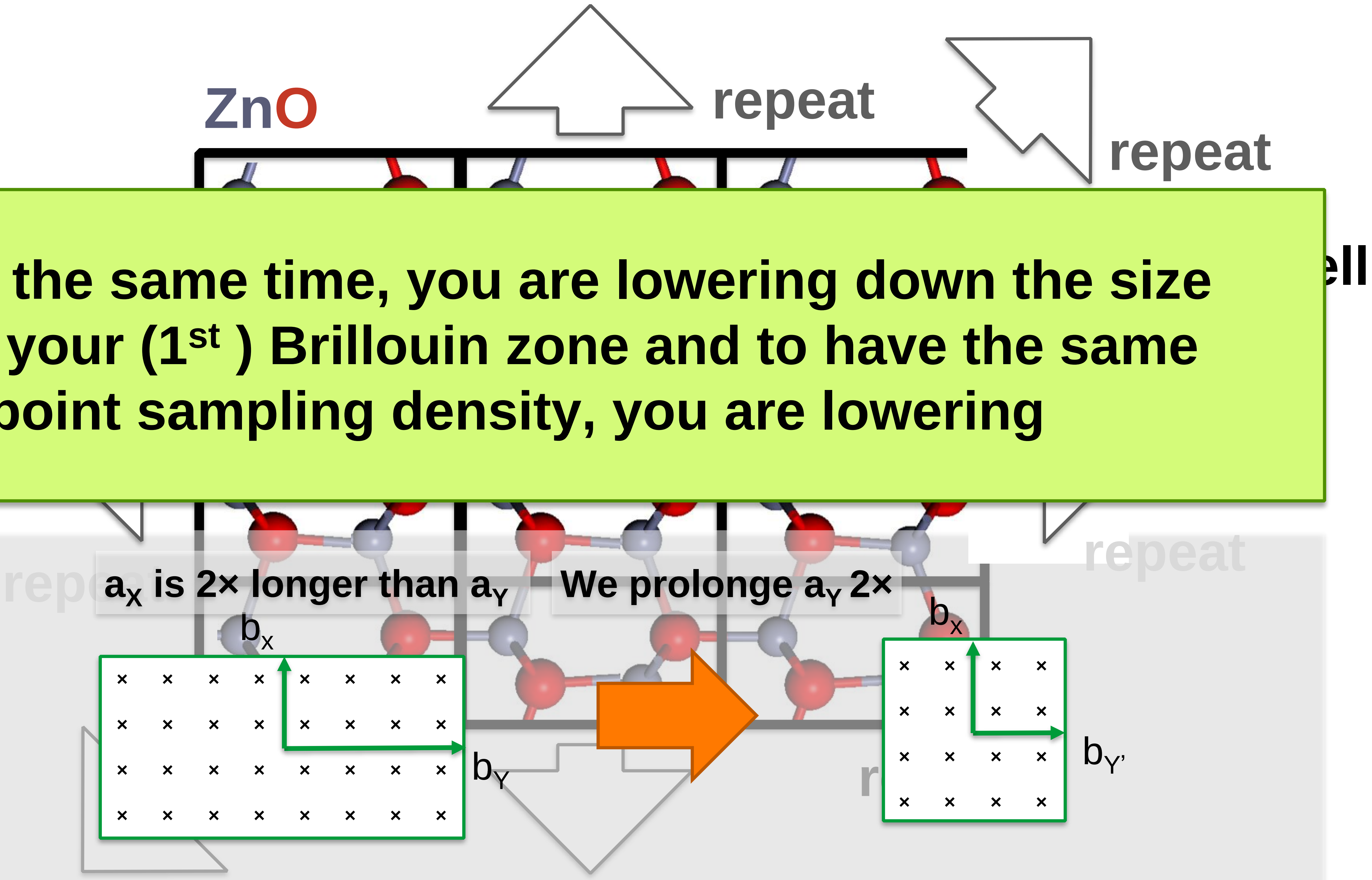
Aalto University
School of Science

Scaling in periodic boundary conditions



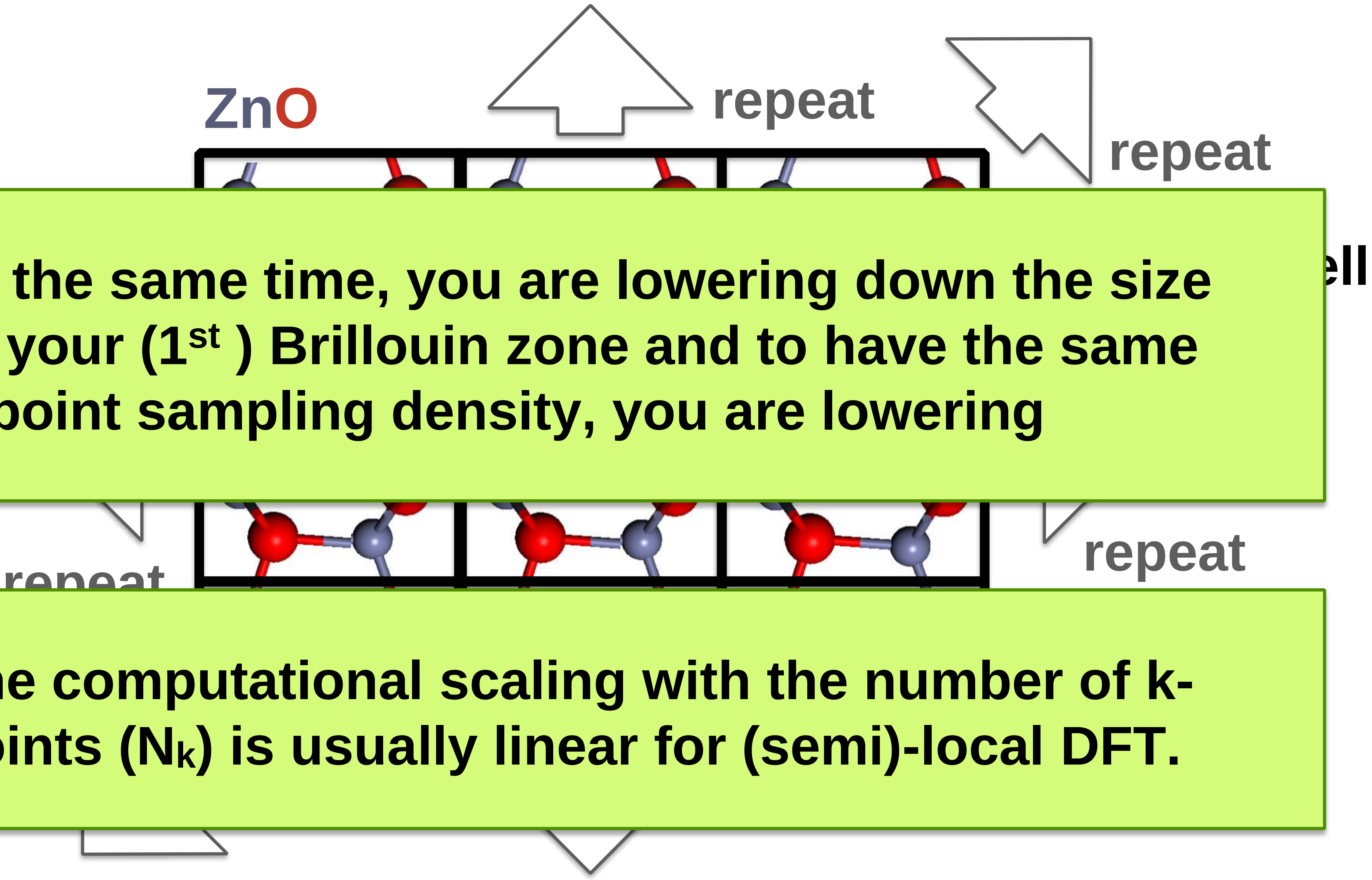
At the same time, you are lowering down the size of your (1st) Brillouin zone and to have the same *k*-point sampling density, you are lowering

Scaling in periodic boundary conditions



At the same time, you are lowering down the size of your (1st) Brillouin zone and to have the same *k*-point sampling density, you are lowering

Scaling in periodic boundary conditions



At the same time, you are lowering down the size of your (1st) Brillouin zone and to have the same *k*-point sampling density, you are lowering

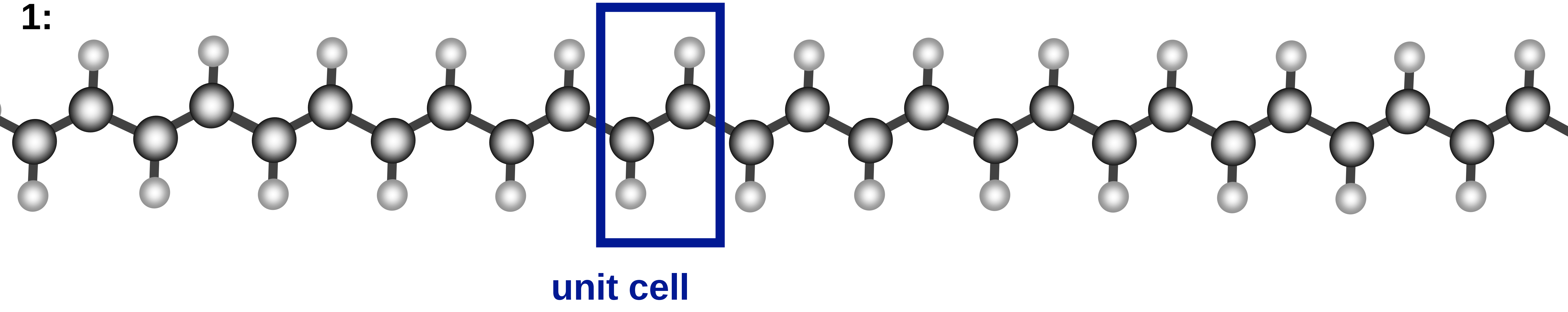
The computational scaling with the number of k-points (N_k) is usually linear for (semi)-local DFT.

Aalto University
School of Science

Scaling exercise 2 - infinite polymer

system	*# atoms in unit cell*	*atomic factor*	*# k-points*	*scaling factor*	*time*
1	*4*	*1*	*4*	*1*	*30s*
2					
3					
4					

1:

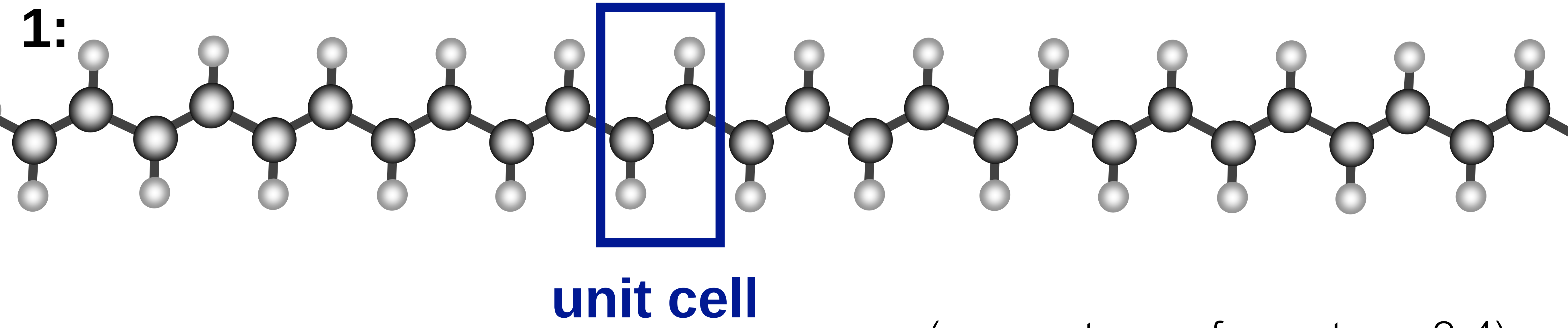


(see handouts for systems 2-4)

Scaling exercise 2 - infinite polymer

system	# atoms in unit cell	atomic factor	# k-points	scaling factor	time
1	4	1	4	1	30s
2	8	2	2	4	2min
3	16	4	1	16	8min
4	32	8	1	128	64min

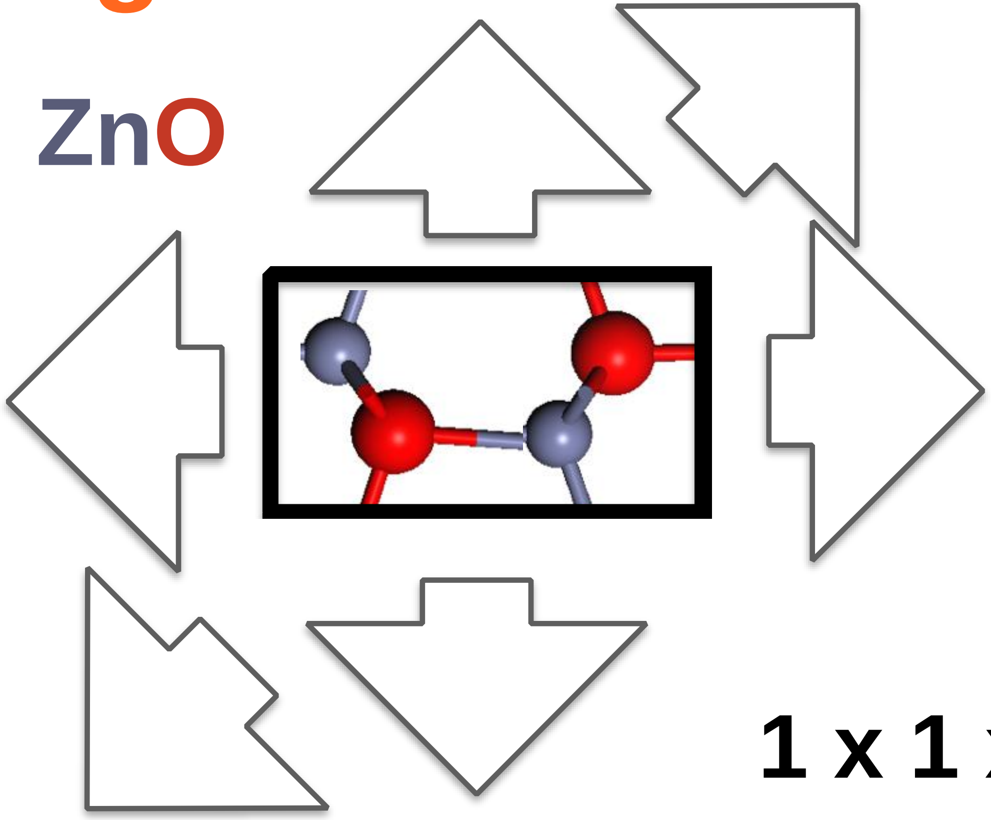
1:



(see next page for systems 2-4)

Scaling in solids

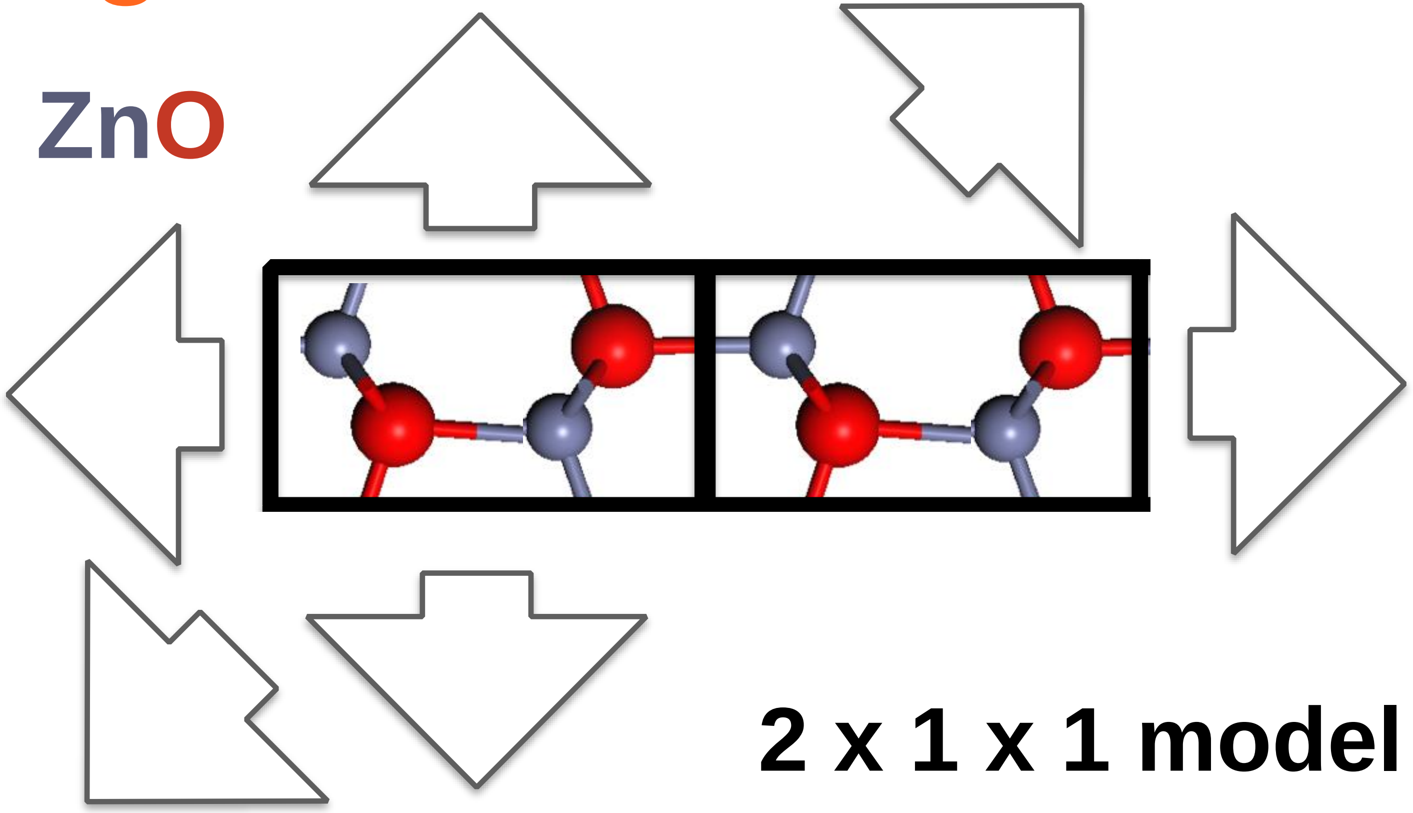
ZnO



1 x 1 x 1 model
8 x 8 x 8 k-points

Time: T_u

Scaling in solids

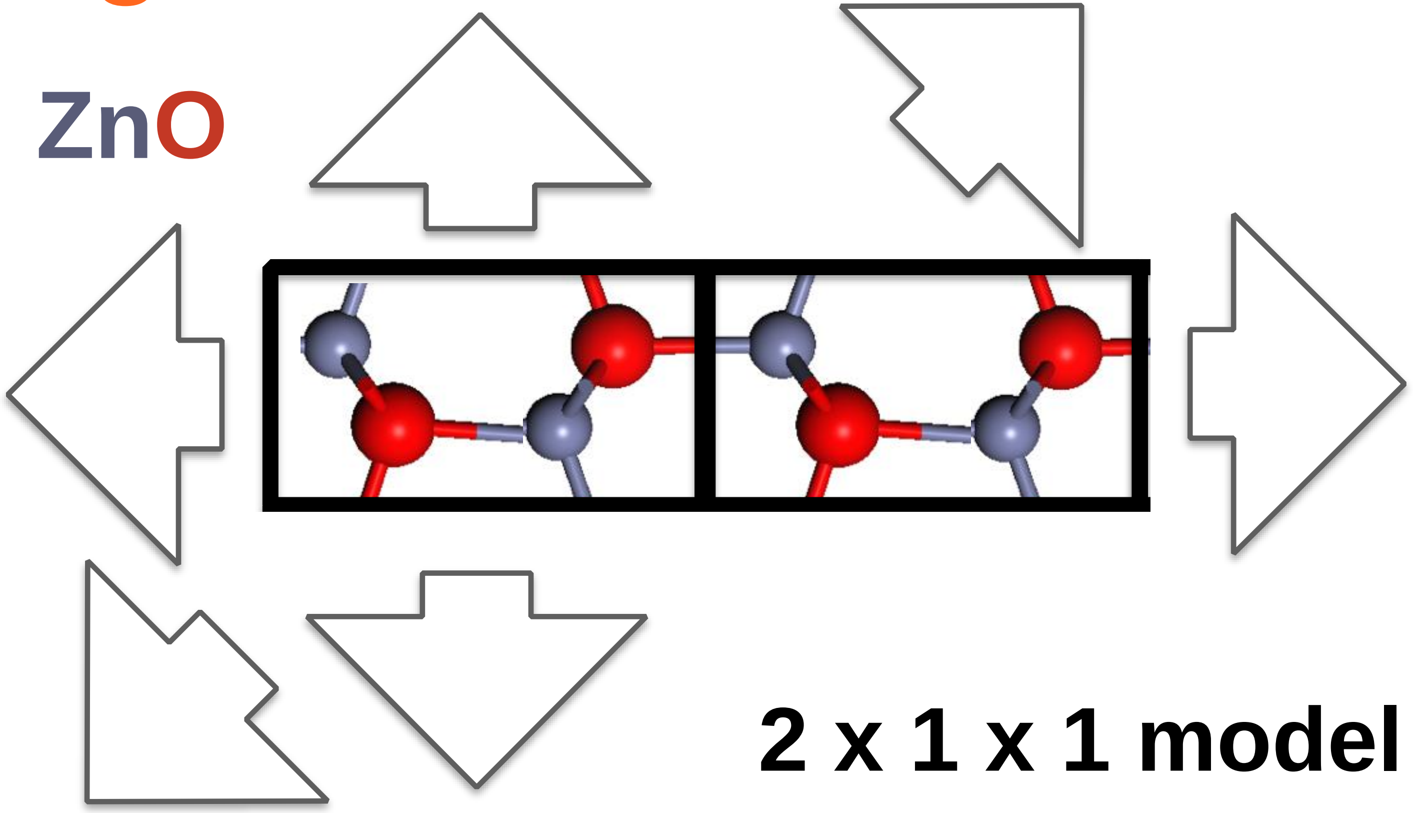


2 x 1 x 1 model
4 x 8 x 8 k-points

The computational scaling with the number of k-points (N_k) is usually linear for (semi)-local DFT.

Scaling in solids

ZnO

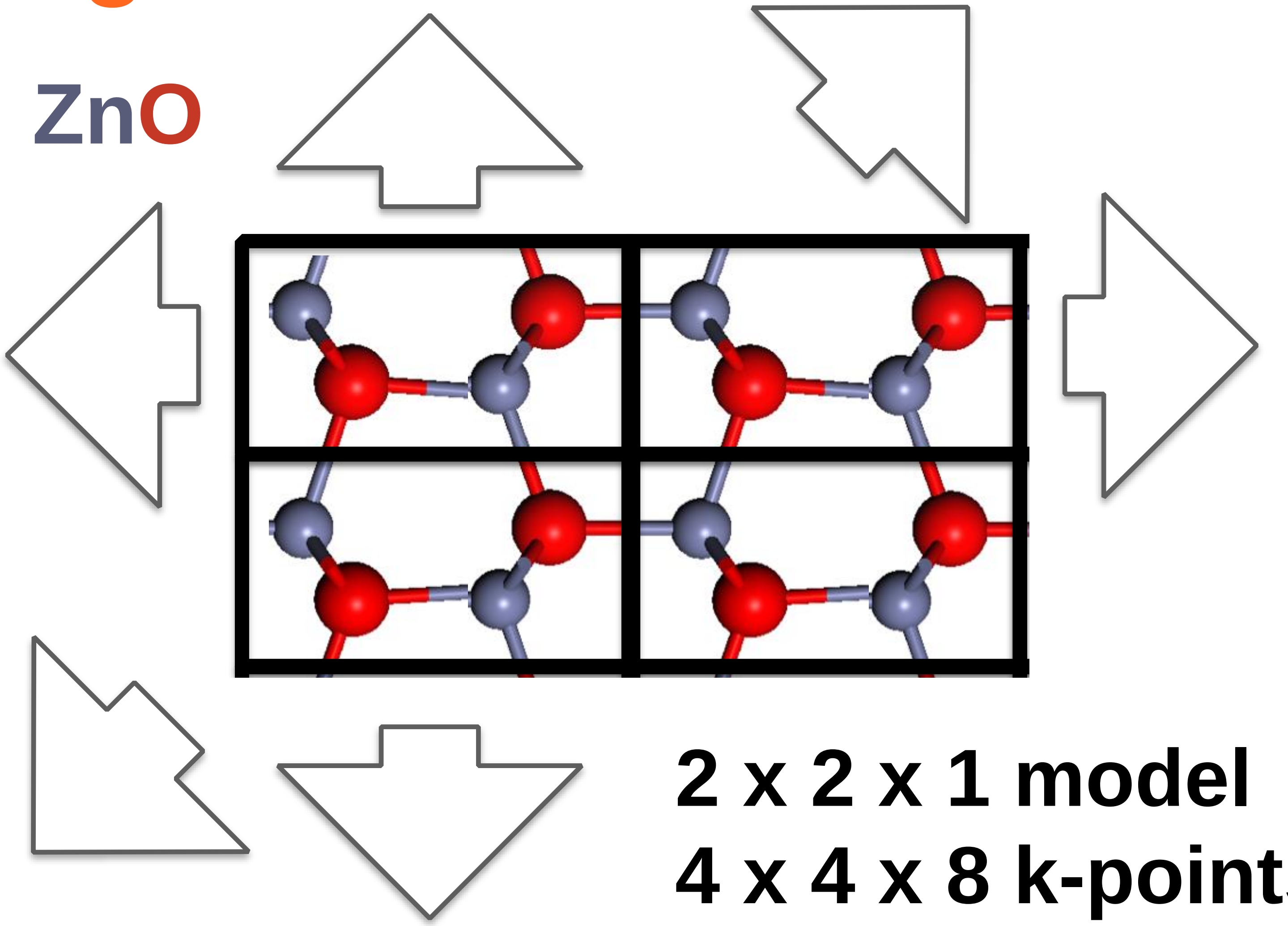


2 x 1 x 1 model
4 x 8 x 8 k-points

Time: $(T_u \times 2^3) \times 1/2 = T_u \times 4$

Scaling in solids

ZnO



2 x 2 x 1 model
4 x 4 x 8 k-points

The k-point benefit continues until you run out of k-points. Then the scaling is cubic in N_a.

Computational budget

The *computational budget* is the total number of cpu hours a project consumes.

Estimating the budget:

- time per calculation (TPC) x number of calculations
- TPC can be estimated based on scaling tests

Scaling tests:

- break large calculations into smaller units
- determine computation time for smallest unit and for successively larger units

Scaling of DFT

- Matrix storage is quadratic in N_b as well as in N_a.
- Matrix inversion is cubic in N_b as well as in N_a.

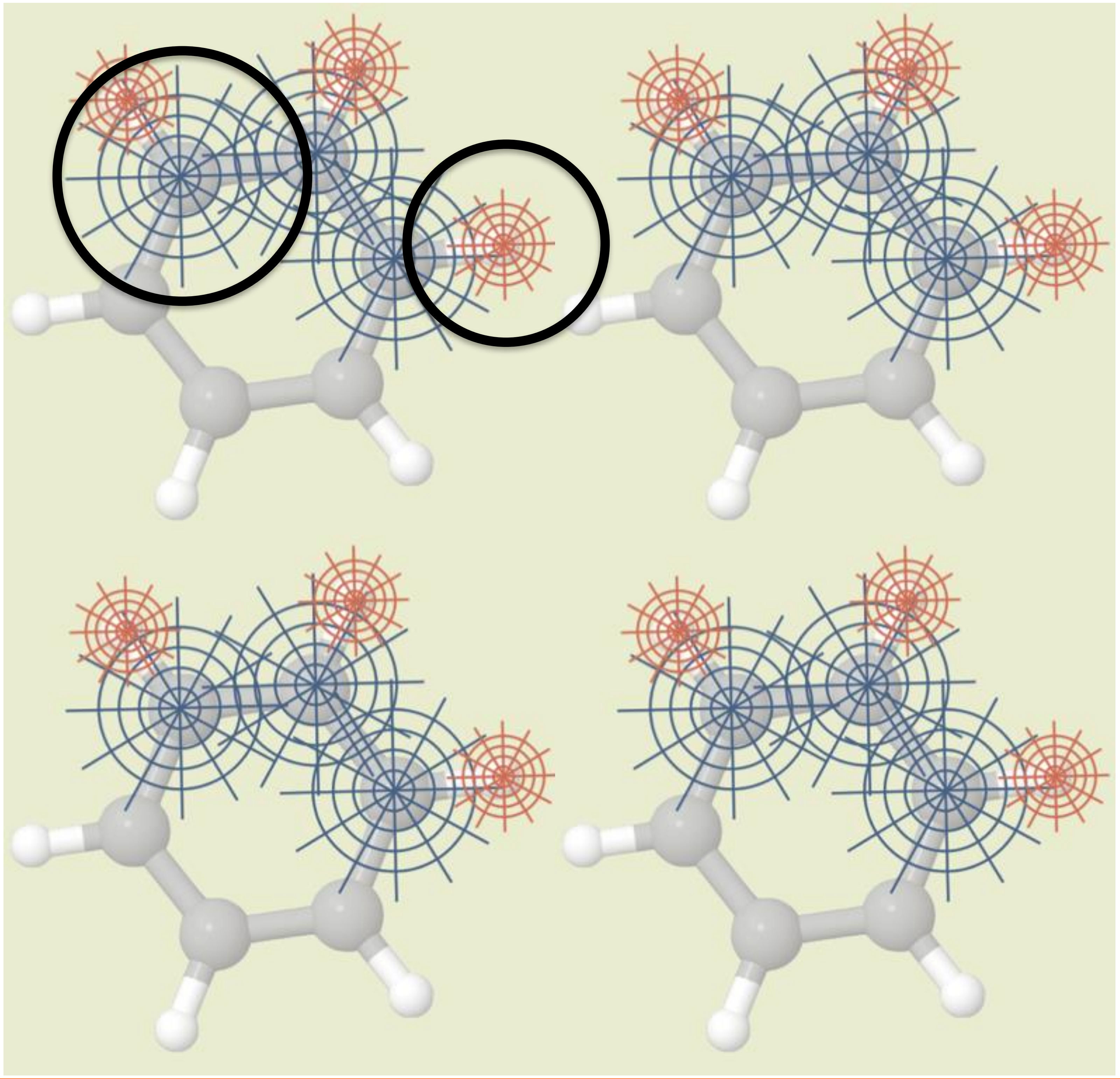
We call this the *formal* scaling of DFT.

If you emphasise the word *formal*, can the actual scaling be better than N_a^3?

$$h_{ij}c_{jn} = \epsilon_n s_{ij} c_{jn}$$

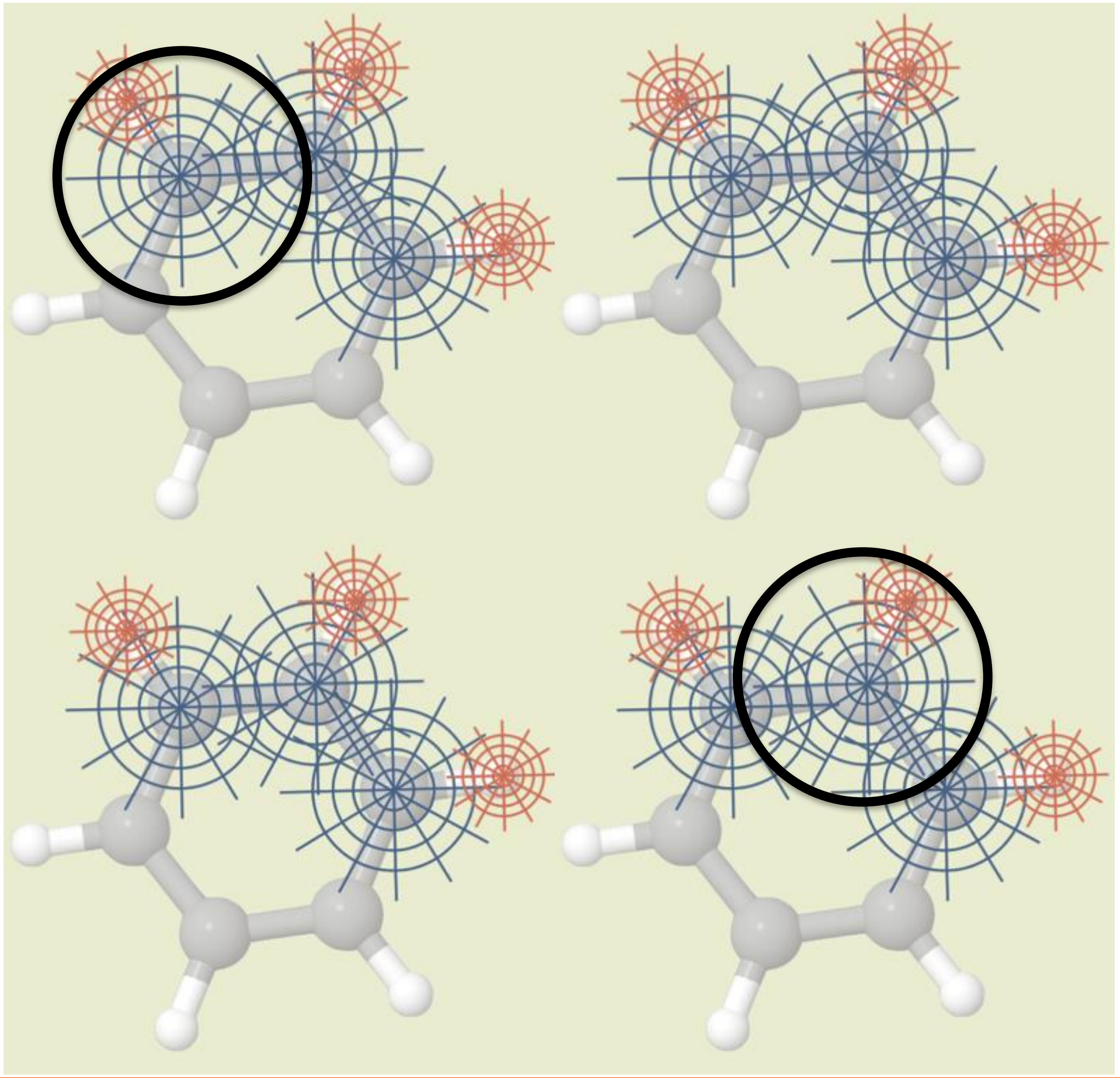
Aalto University
School of Science

Locality of the basis functions



$$s_{ij} = 0$$

Locality of the basis functions



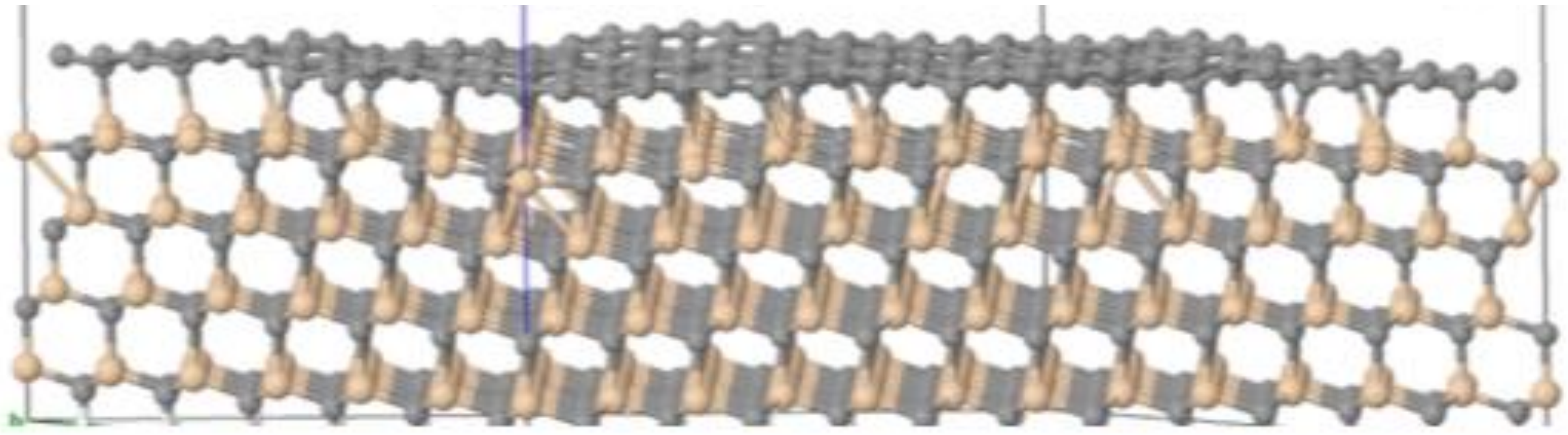
$$s_{ij} = 0$$

$$h_{ij} \ll 1$$

Interactions between far away basis functions become less important.

Scalability: benchmark system

Graphene on SiC:



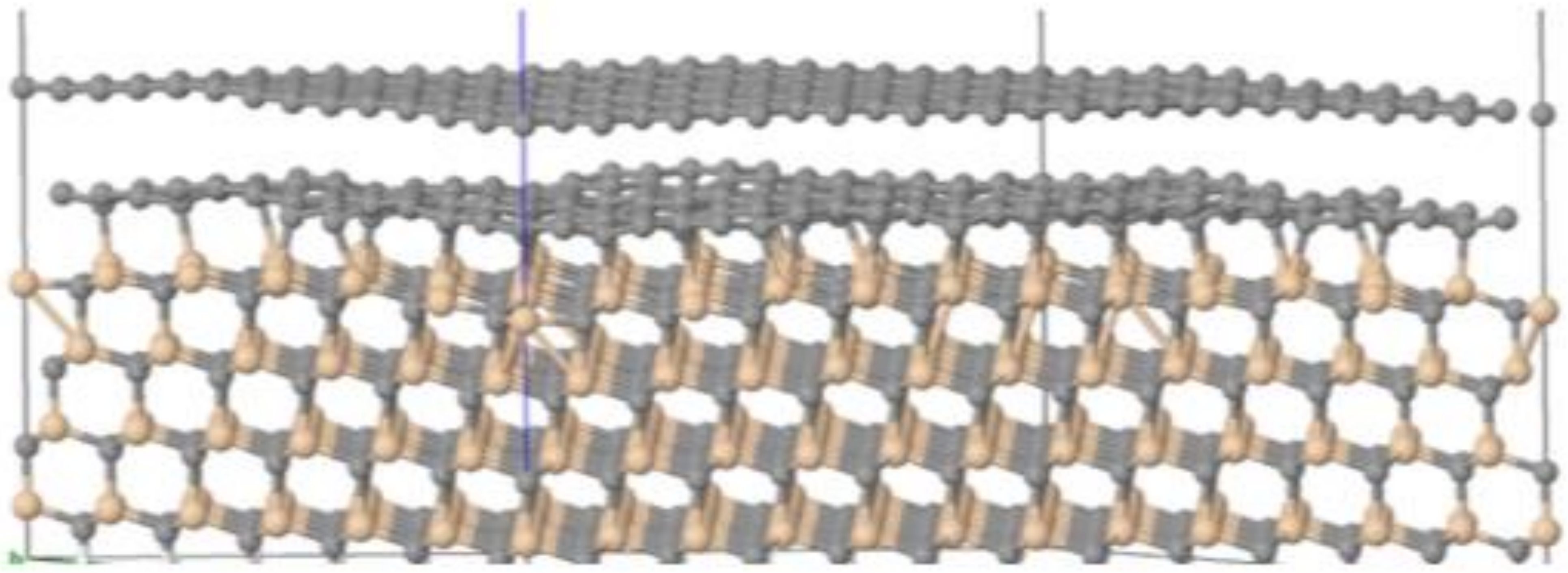
Zero layer graphene: ZLG (1310 atoms in unit cell)

Aalto University
School of Science

Benchmark: Björn Lange, Duke University 2014

Scalability: benchmark system

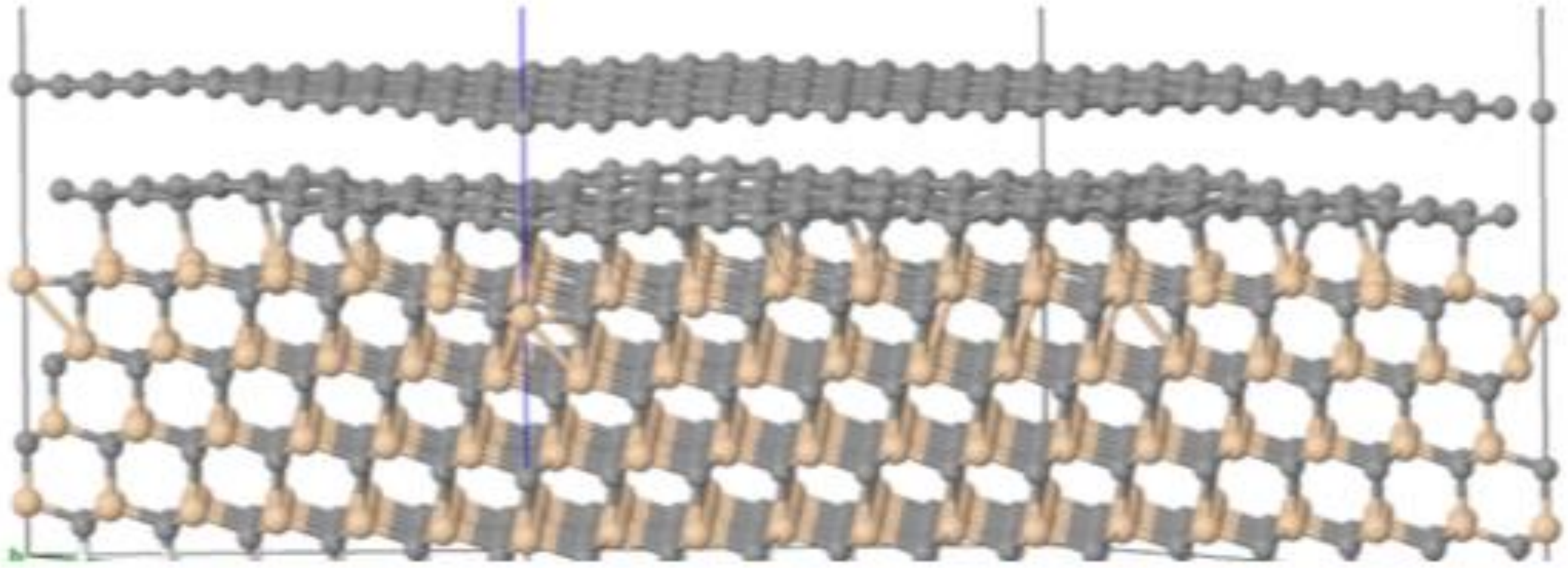
Graphene on SiC:



Mono layer graphene: ZLG (1648 atoms in unit cell)

Scalability: benchmark system

Graphene on SiC:

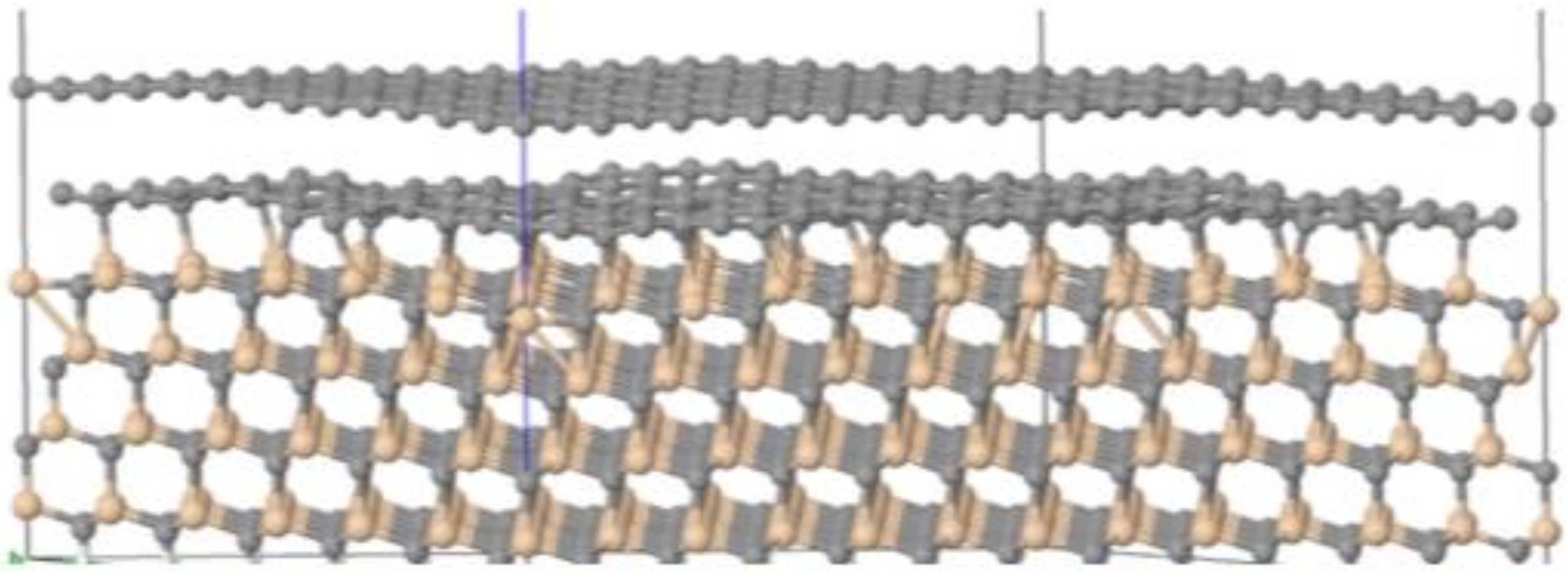


keep growing graphene

Double layer graphene: DLG (1986 atoms in unit cell)

Scalability: benchmark system

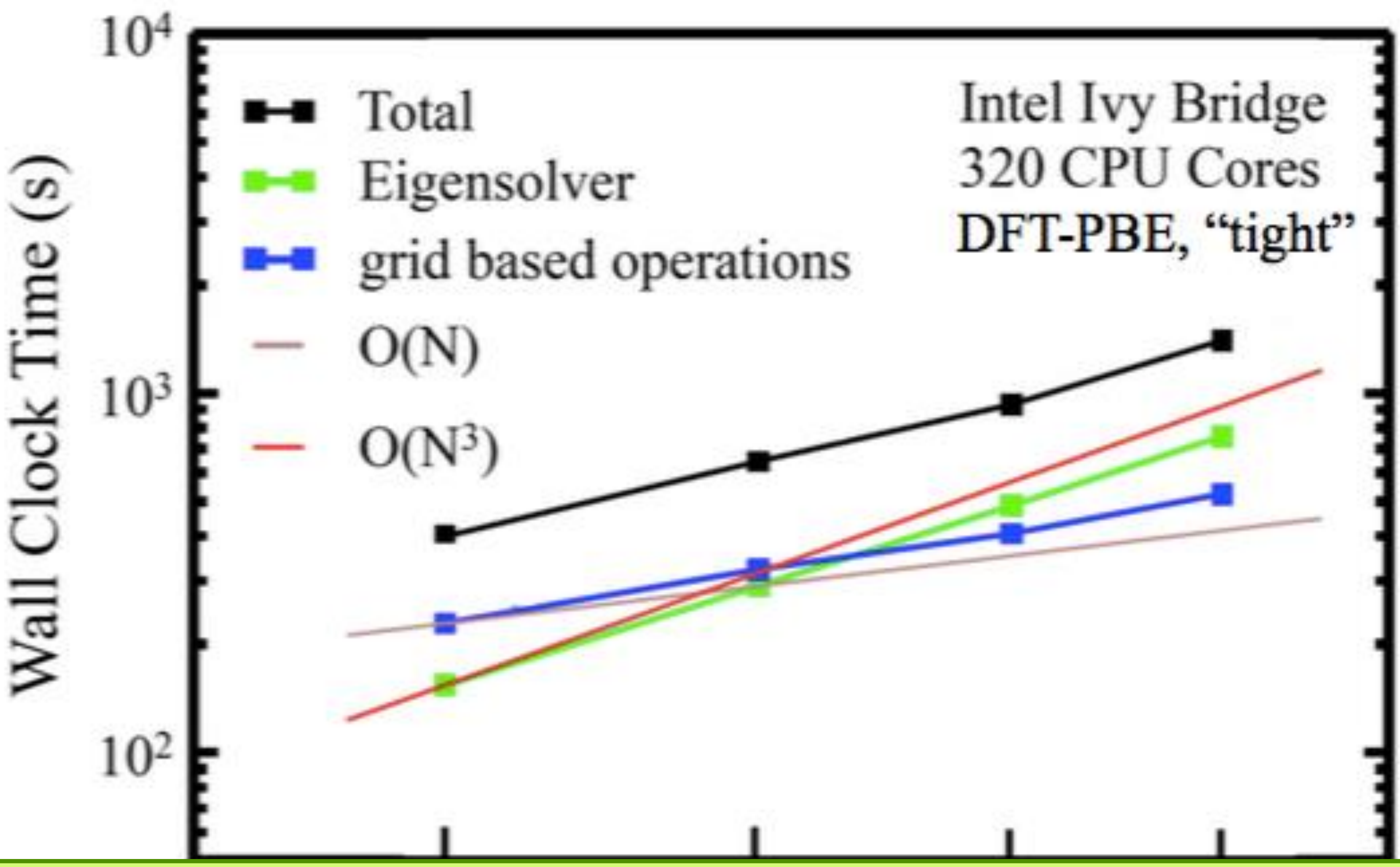
Graphene on SiC:



Triple layer graphene: TLG (2324 atoms in unit cell)

Benchmark: Björn Lange, Duke University 2014

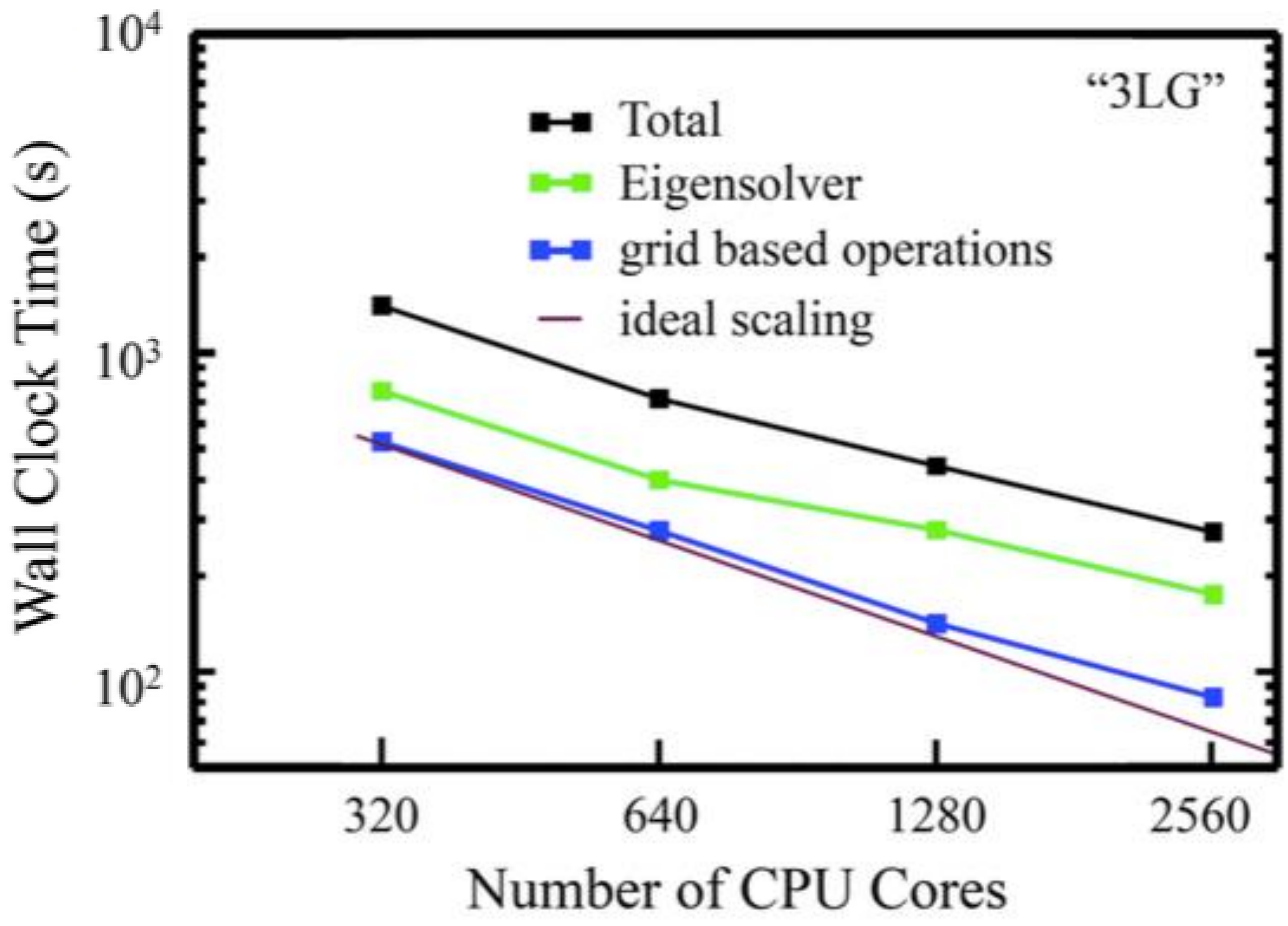
Scalability of local and semi-local DFT



Actual performance is better than cubic, because the eigenvalue solver does not dominate.

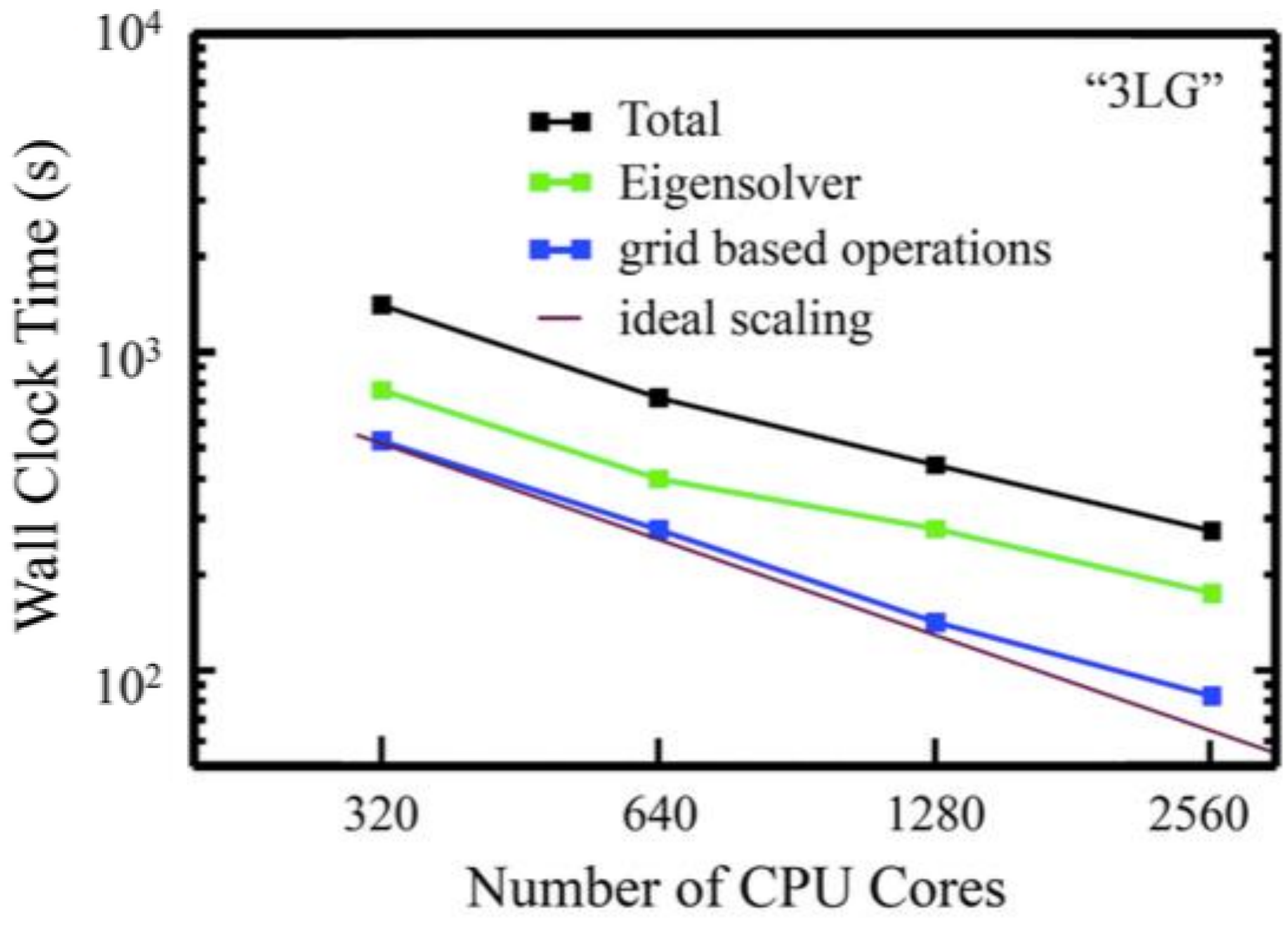
Benchmark: Björn Lange, Duke University 2014

Scalability of local and semi-local DFT



Benchmark: Björn Lange, Duke University 2014

Scalability of local and semi-local DFT



You can have max. 4-8 cores on the local desktops
On the Aalto comp. cluster Triton, you can go up to 192 cores.
If you want to use Triton for the project, but did not an access, please contact us soon!

Aalto University
School of Science

Benchmark: Björn Lange, Duke University 2014

Questions?

Before doing anything please run in the terminal:

pip3 install phonopy   (--user)

Enjoy our "Phonon calculations" tutorial:

- **Download from Mycourses**
- **/work/courses/unix/PHYS/E0546/TUTORIALS**
- **Ask for printouts**

Don't forget to adjust the number of k-points, once you are changing the size of your supercell!

Interesting links related to the tutorial:

https://phonopy.github.io/phonopy/workflow.html

https://www.tcm.phy.cam.ac.uk/~jry20/gipaw/tutorial_vib.pdf

And also FHI-aims manual.

Don't forget to put your name in the list of attendance!

Aalto University
School of Science