

First principles simulation of Quantum Materials

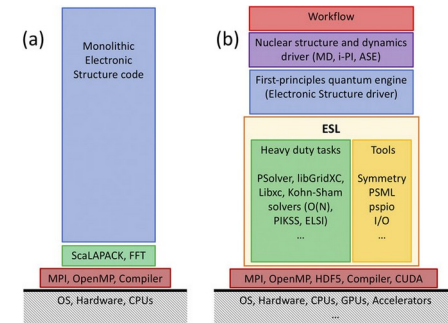
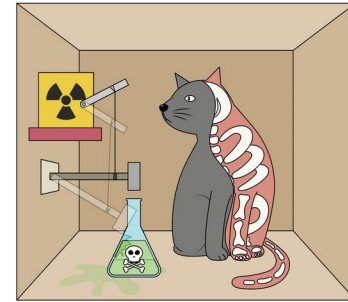
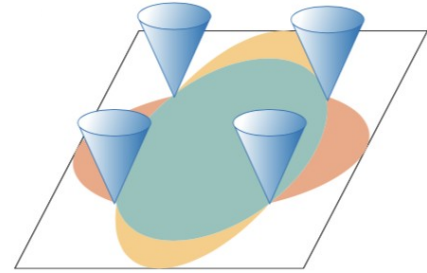
A.H. Kole, A.R. Botello-Méndez, Z. Zanolli
Debye Institute for Nanomaterials Science, Utrecht University
a.h.kole@uu.nl

SURF Advanced Computing User Day 2024

Overview



- 1) Quantum Materials
- 2) First-principles simulations
- 3) High-performance computing
- 4) Good practices
- 5) Future



Quantum Materials

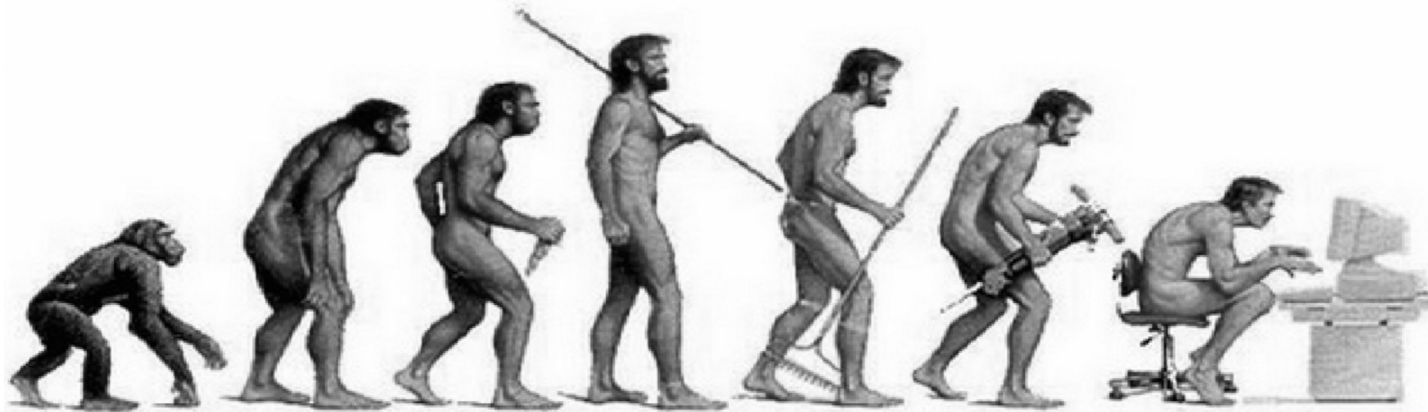
What drives human development?



Major advances in human civilization are driven by materials:

stone, bronze, iron...

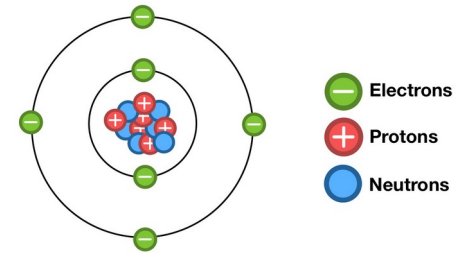
... silicon age



But... The silicon era is approaching its limits.



The Atom



All materials are quantum

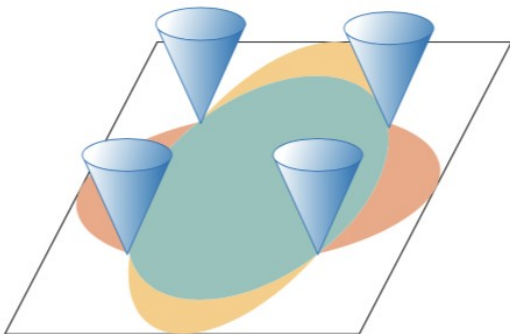
But some materials are more quantum than others.

- Zeila Zanolli (2020)

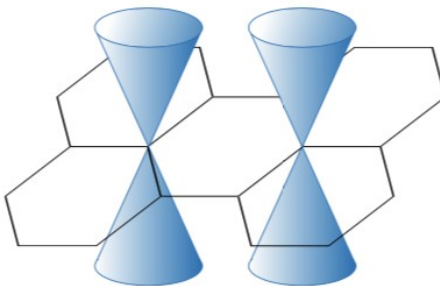
Quantum Materials



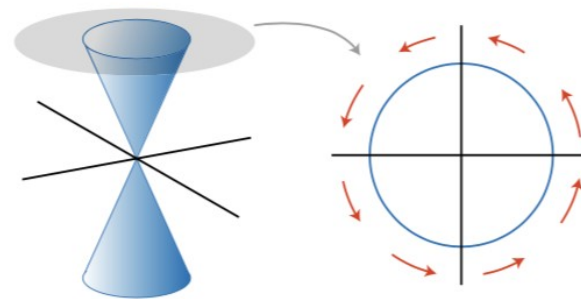
Superconductors



Dirac Materials



Topological insulators



Keimer & Moore Nat Phys 2017

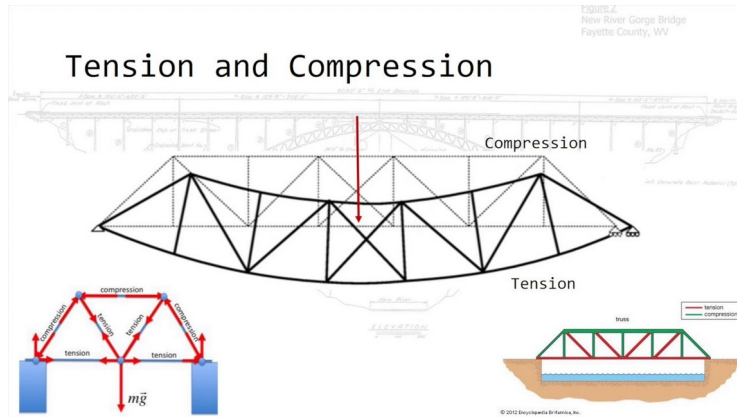
**Quantum behaviour at
macroscopic scales**



Quantum technology:
Low-dissipation electronics
Photovoltaics
Quantum information

Using simulations to guide quantum materials design

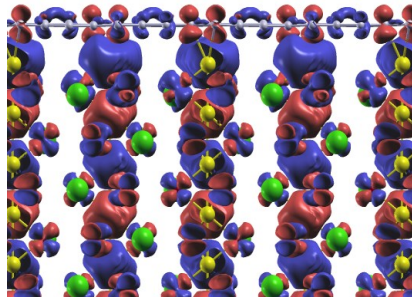
- Perfect atomic control
- Try before you buy/fabricate
- Easy exploration of many different materials/combinations
- Close collaboration with experimentalists



He tested thousands and thousands of other materials to use for the filament.

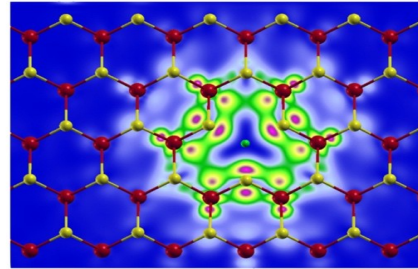
The Franklin Institute on Thomas Edison's search for the lightbulb filament
<https://fi.edu/en/science-and-education/collection/edisons-lightbulb>





Topological Matter

- Low-energy consumption electronics

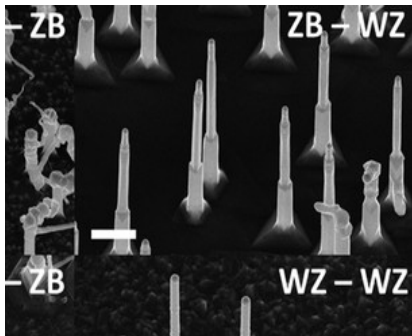


Theoretical Spectroscopy

- Quantum computing
- Cryptography
- Quantum sensing
- Photovoltaics

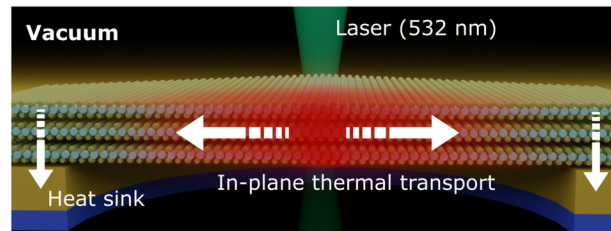
Superconductivity

- Q computing, Qubits



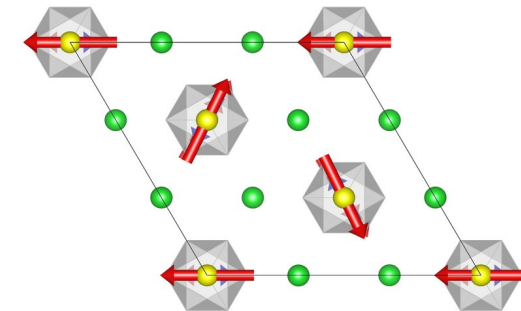
Quantum Electron & Thermal Transport

- Semiconductor industry



Magnetism

- Spin Qubits

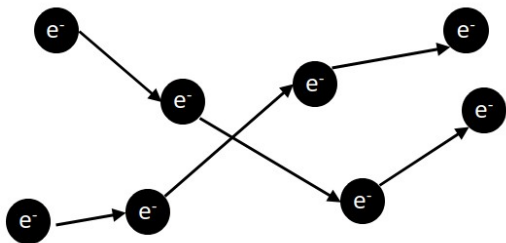


Quantum Materials

Normal metals vs superconductors



Normal metal



Electrons repel each other (Coulomb)

Electron-electron scattering
→ finite resistance

Superconductor



Effective attractive electron-electron interaction

- Formation of Cooper pairs
- Collective motion without scattering

Normal metals vs superconductors



Superconductor



Effective attractive electron-electron interaction

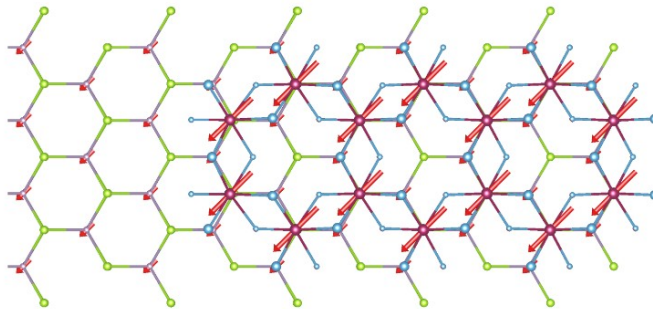
- Formation of Cooper pairs
- Collective motion without scattering

Interplay superconductivity and magnetism



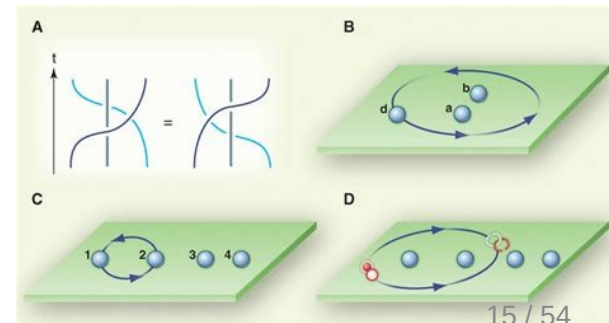
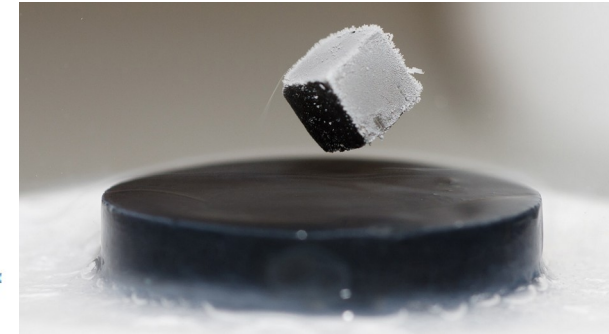
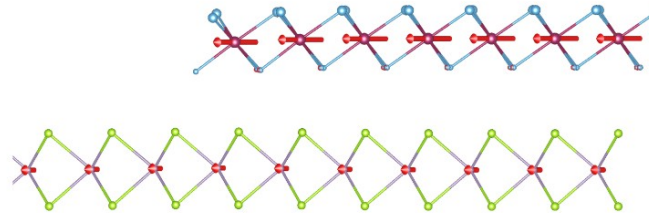
- Conventional superconductors expel magnetic fields (Meissner effect)
- Heterostructures of magnetic atoms and superconductors yield exciting new physics
- Potential applications in quantum computing

a)



Cuperus submitted 2024

b)



First-principles simulations

First-principles



Definition of 'first principle'



first principle

in British English



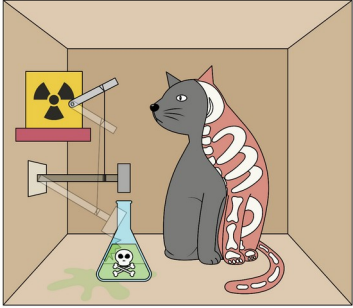
noun (usually plural)

1. one of the fundamental assumptions on which a particular theory or procedure is thought to be based
2. an axiom of a mathematical or scientific theory

Collins English Dictionary. Copyright © HarperCollins Publishers

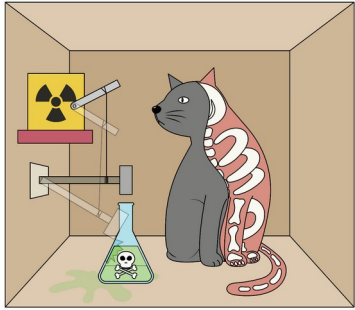
Computing the properties of the system in a **parameter-free** way, including all the material specific details

Quantum description

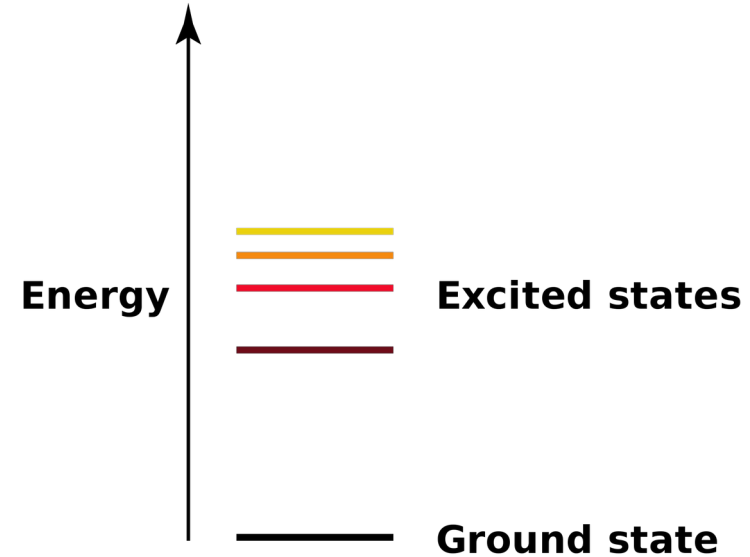
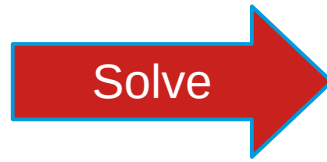


$$H\Psi_n = E_n\Psi_n$$

Quantum description

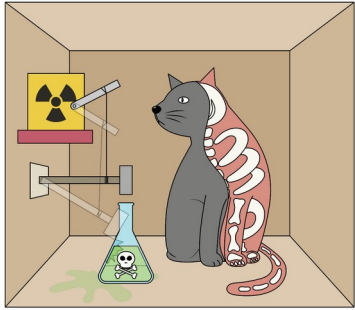


$$H\Psi_n = E_n\Psi_n$$



Quantized states Ψ_n with energy E_n

Quantum description



Ground state properties

- Electronic
- Structural
- Magnetic
- Topological

Excited state properties

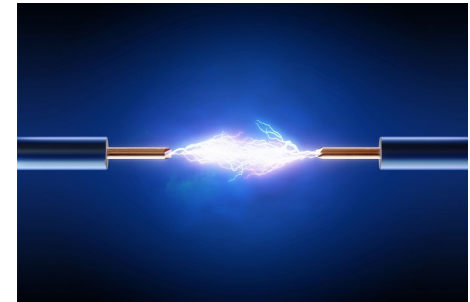
- Optical properties

Time-dependent properties

- Transport

$$H\Psi_n = E_n\Psi_n$$

Solve

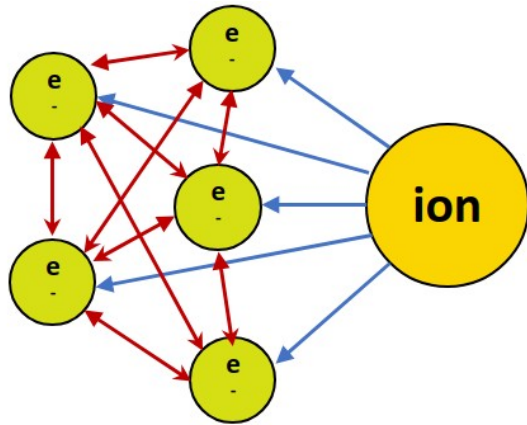


Overview of computational approach



SC: Superconductivity

Density Functional Theory

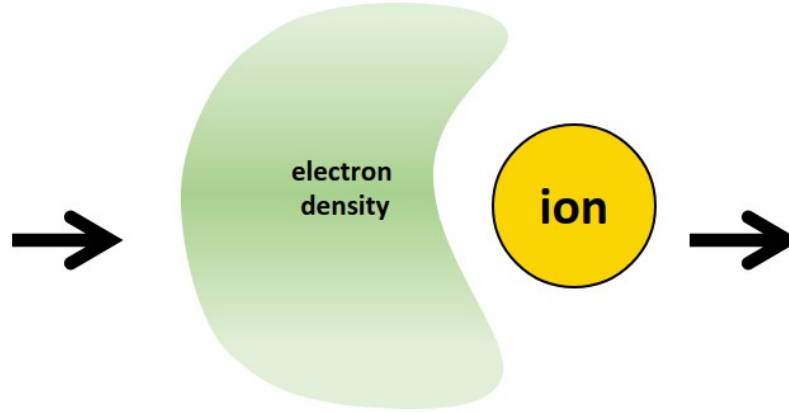


Many-Body

N Interacting particles

$$\psi(r_1, s_1; r_2, s_2; \dots; r_N, s_N)$$

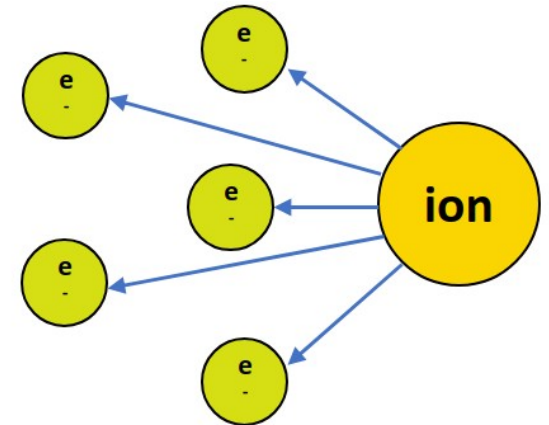
$$H\psi = (T + V + U)\psi = E\psi$$



Hohenberg-Kohn

Energy is a unique functional of **ground state** electron density n

$$E = E[n(r)]$$



Kohn-Sham

N Independent particles auxiliary potential and same **ground state** electronic density

$$\sim \varphi_i(x, s)$$

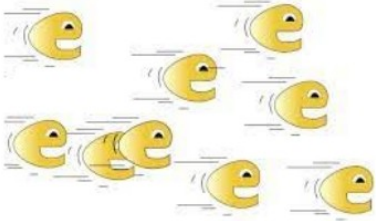
$$H\varphi = (T - E_F + V_{eff})\varphi = E\varphi$$

$$V_{eff}(\mathbf{x})[\rho] = V_{ext}(\mathbf{x}) + \int \frac{\rho(\mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|} d\mathbf{x}' + \frac{\delta E_{xc}[\rho]}{\delta \rho(\mathbf{x})}$$

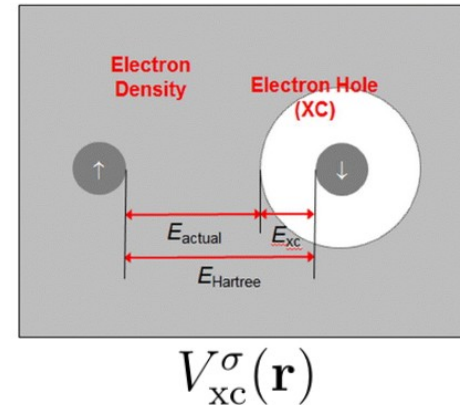
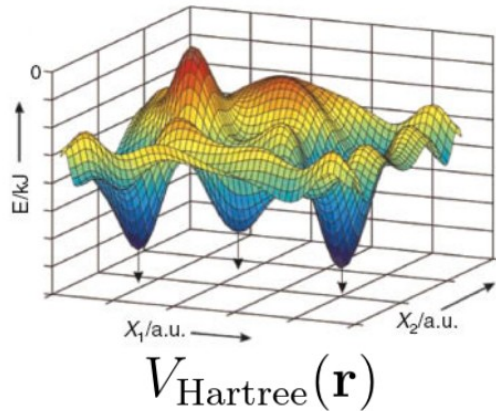
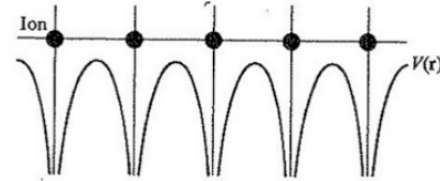
Hamiltonian



$$-\frac{1}{2}\nabla^2$$



$$V_{\text{ext}}(\mathbf{r})$$



Basis Sets



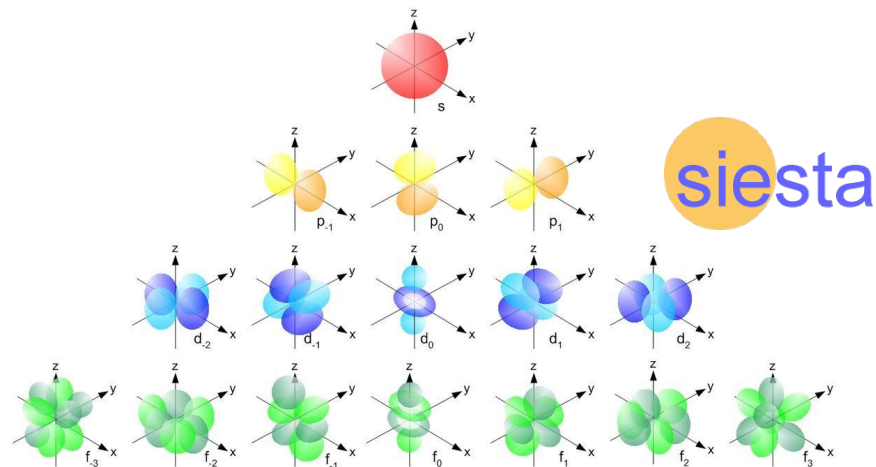
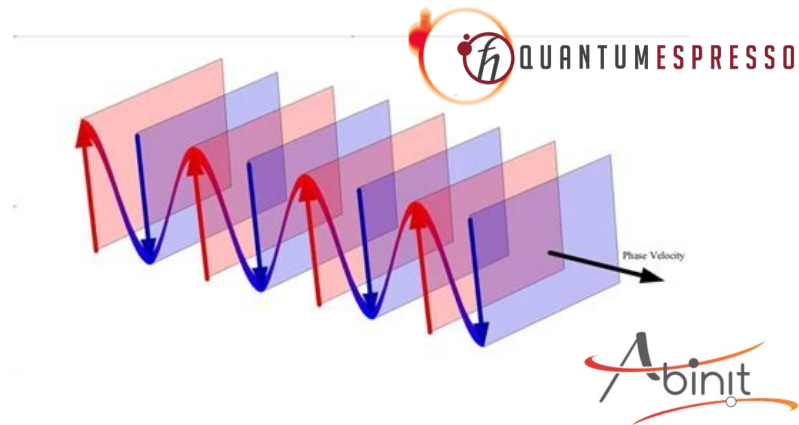
How to represent the wavefunction?

- On a regular grid => impractical
- Better: expand in terms of known functions

$$\psi = \sum_i c_i \phi_i$$

$$H_{ij} = \int \phi_i(r) H(r) \phi_j(r) dr$$

$$H C_i = E_i C_i$$





$$H_{\text{SC}} = H_{\text{DFT}} + \Delta$$



Our implementation in SIESTA

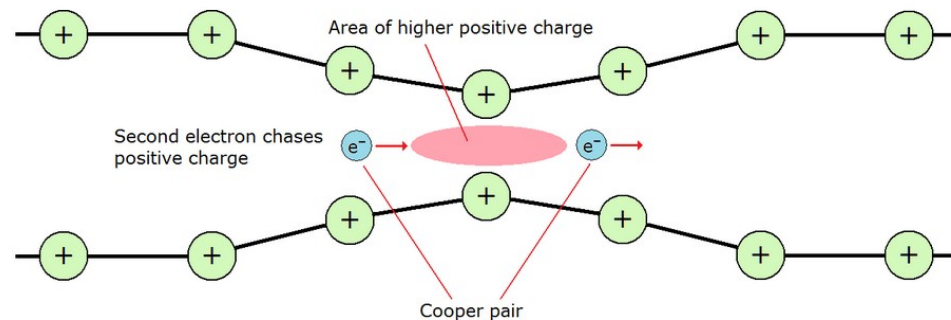
Reho, Wittemeier, Kole, Ordejón, Zanolli
Phys. Rev. B 110, 134505 (2024)



First-principles superconductivity



$$H_{\text{SC}} = H_{\text{DFT}} + \Delta$$



Our implementation in SIESTA

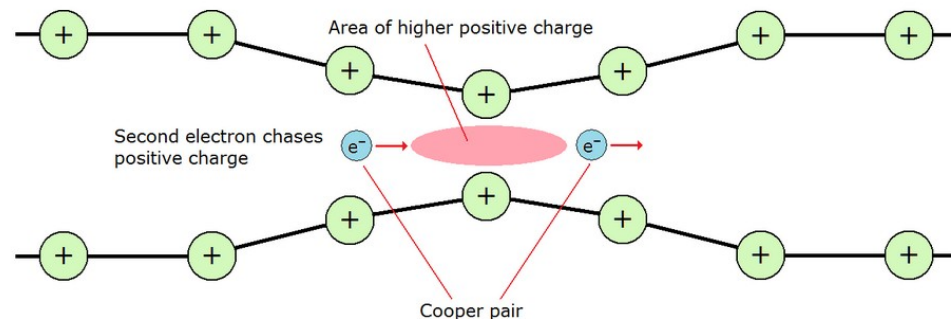
Reho, Wittemeier, Kole, Ordejón, Zanolli
Phys. Rev. B 110, 134505 (2024)



First-principles superconductivity



$$H_{\text{SC}} = H_{\text{DFT}} + \Delta$$



Our implementation in SIESTA

Reho, Wittemeier, Kole, Ordejón, Zanolli
Phys. Rev. B 110, 134505 (2024)



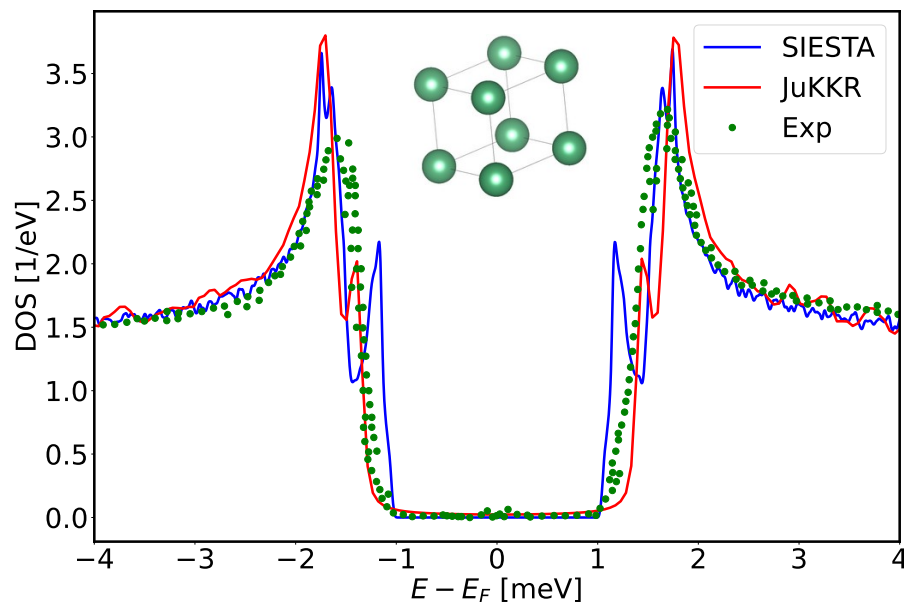
$$\begin{pmatrix} H_{\text{DFT}} & \Delta \\ -(\Delta)^* & - (H_{\text{DFT}})^* \end{pmatrix} \begin{pmatrix} u_i^\sigma \\ v_i^\sigma \end{pmatrix} = \epsilon_i \begin{pmatrix} u_i^\sigma \\ v_i^\sigma \end{pmatrix}$$

u_i^σ electrons
 v_i^σ holes

First-principles superconductivity



siesta



Reho, Wittemeier, Kole, Ordejón, Zanolli
Phys. Rev. B 110, 134505 (2024)



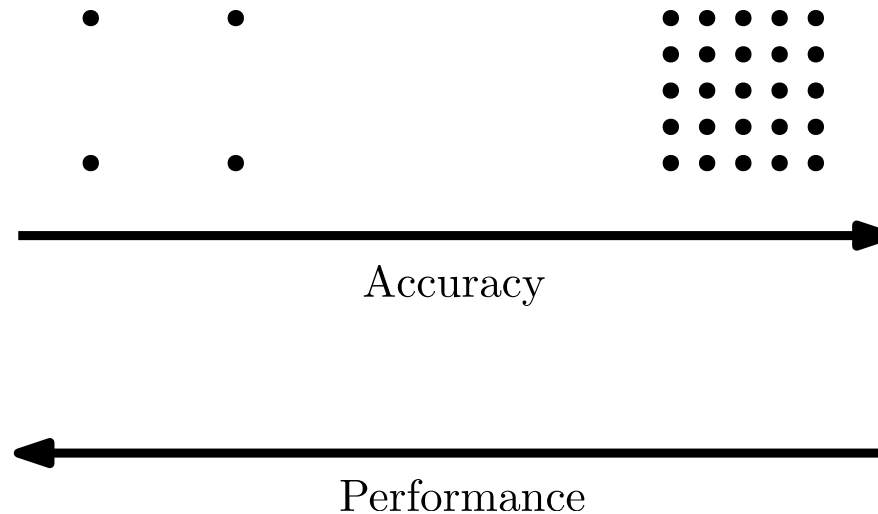
Example for bulk niobium

- Parameter-free except for superconducting interaction
- Direct comparison with experiment
- Measurable properties are material specific
- Small energy scales require high accuracy
- Valid for heterostructures

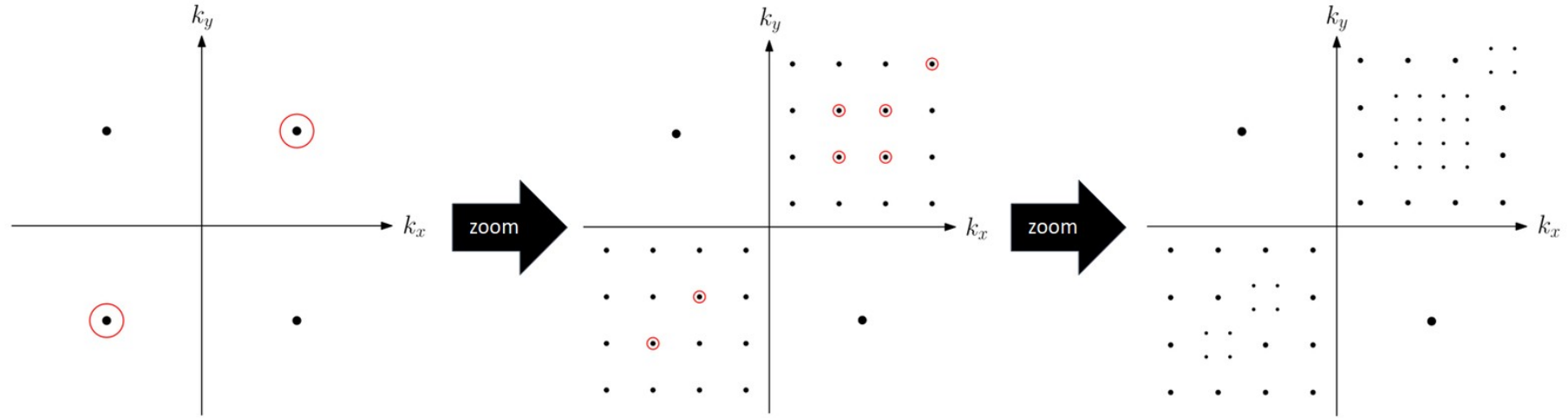
Performance vs Accuracy



- Size of basis set, number of grid points for integration, etc.
- Higher accuracy means more expensive calculation



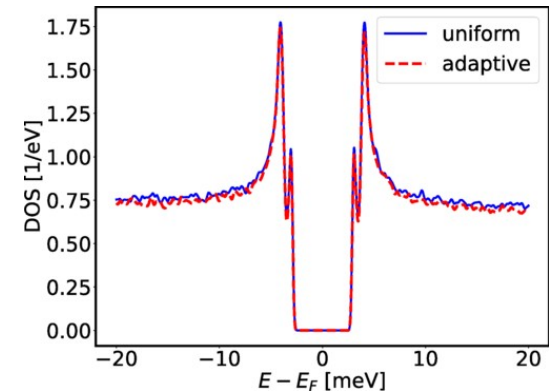
Adaptive sampling



Reho, Wittemeier, Kole, Ordejón, Zanolli
Phys. Rev. B 110, 134505 (2024)



a.h.kole@uu.nl



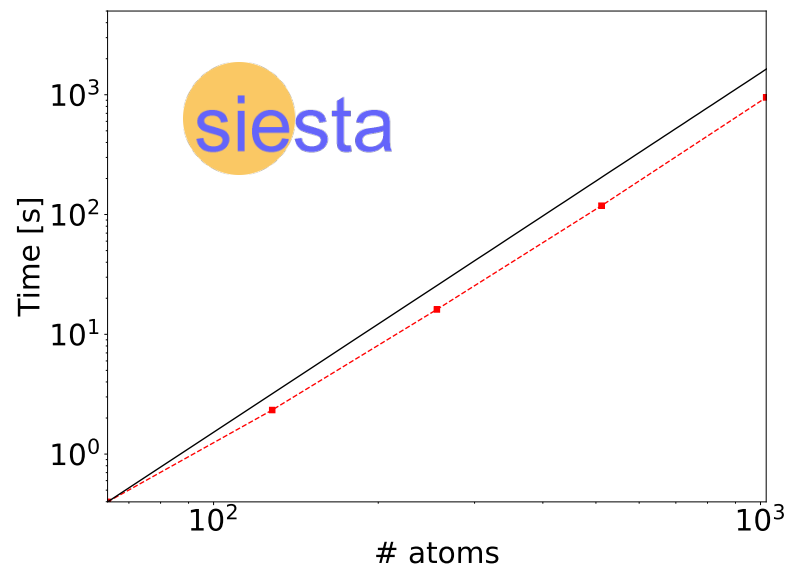
High-performance computing

Why we need HPC

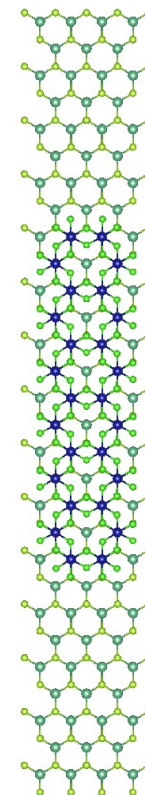


Eigenvalue problem

$$\text{time} \propto N_{\text{atom}}^3$$



Heterostructures require large simulations



Cuperus submitted 2024

Parallelization strategies



Different levels of parallelization

Parallelization strategies



Different levels of parallelization

- 1) Distributed parallel linear algebra (i.e. ScaLAPACK, ELPA)

3x3 process grid

P1	P2	P3
P4	P5	P6
P7	P8	P9

10x9 matrix

0	1	2	3	4	5	6	7	8
10	11	12	13	14	15	16	17	18
20	21	22	23	24	25	26	27	28
30	31	32	33	34	35	36	37	38
40	41	42	43	44	45	46	47	48
50	51	52	53	54	55	56	57	58
60	61	62	63	64	65	66	67	68
70	71	72	73	74	75	76	77	78
80	81	82	83	84	85	86	87	88
90	91	92	93	94	95	96	97	98

P1

0	1	6	7
10	11	16	17
60	61	66	67
70	71	76	77

P2

2	3	8
12	13	18
62	63	68
72	73	78

P3

4	5
14	15
64	65
74	75

P4

20	21	26	27
30	31	36	37
80	81	86	87
90	91	96	97

P5

22	23	28
32	33	38
82	83	88
92	93	98

P6

24	25
34	35
84	85
94	95

P7

40	41	46	47
50	51	56	57

P8

42	43	48
52	53	58

P9

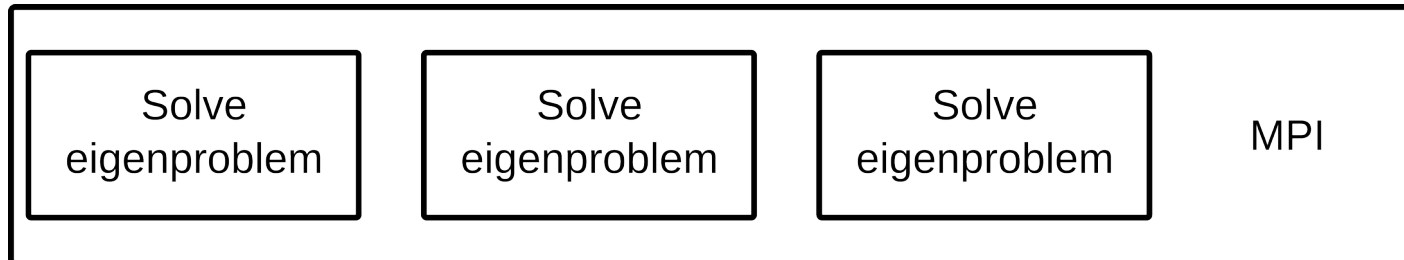
44	45
54	55

Parallelization strategies



Different levels of parallelization

- 1) Distributed parallel linear algebra (i.e. ScaLAPACK, ELPA)
- 2) Solving more eigenvalue problems simultaneously

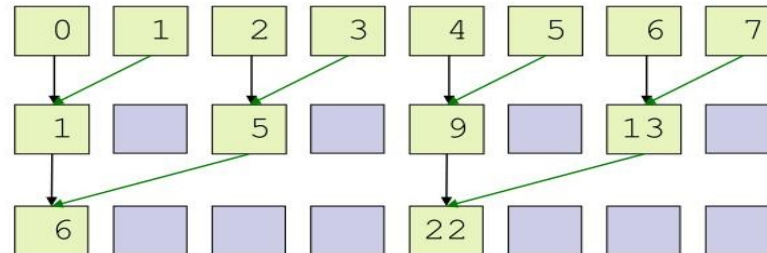


Parallelization strategies

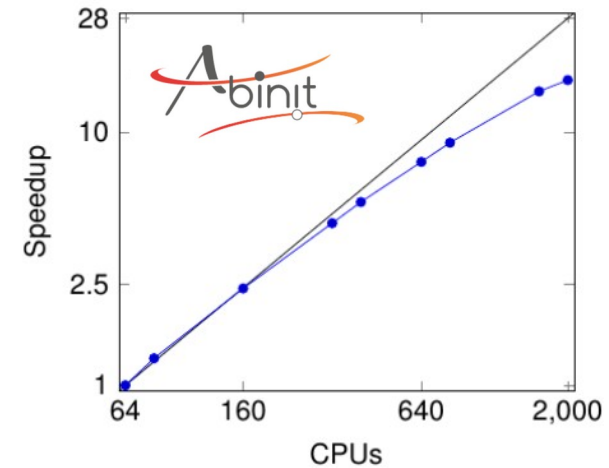
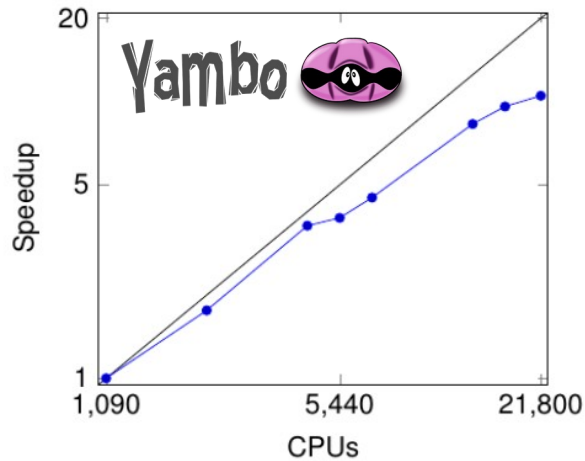
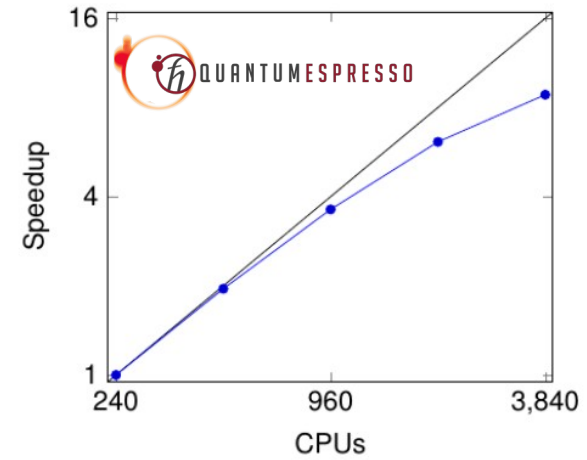
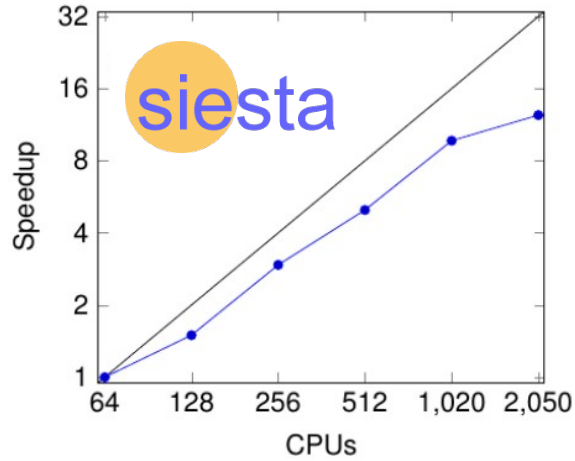


Different levels of parallelization

- 1) Distributed parallel linear algebra (i.e. ScaLAPACK, ELPA)
- 2) Solving more eigenvalue problems simultaneously
- 3) Parallel integration



Parallelization strategies

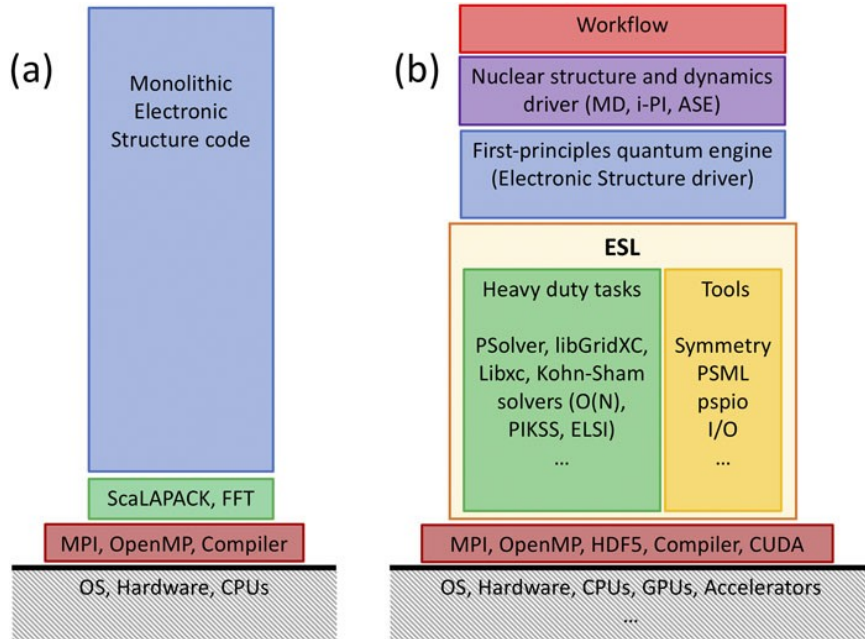


Good practices

Modular code design



Electronic Structure Library



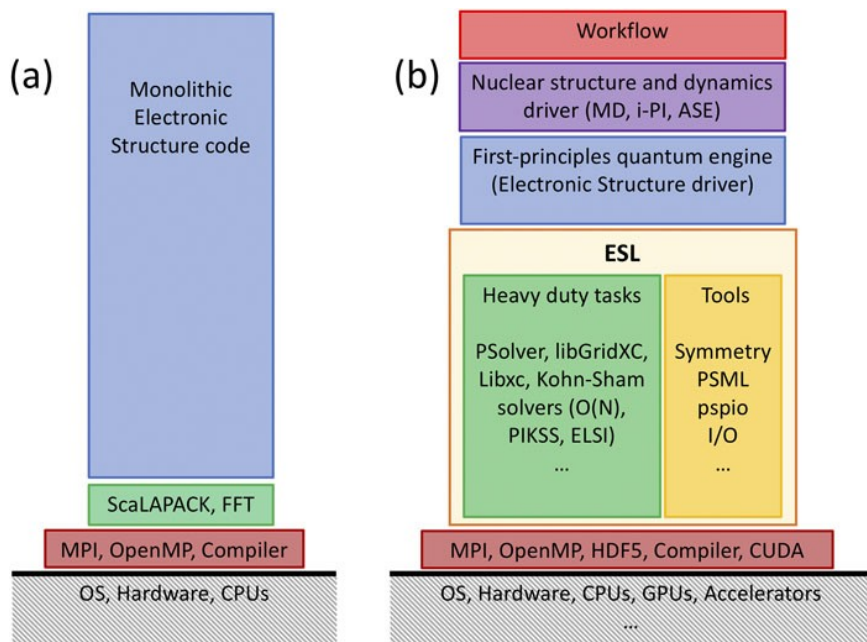
J. Chem. Phys.. 2020; 153(2). doi:10.1063/5.0012901



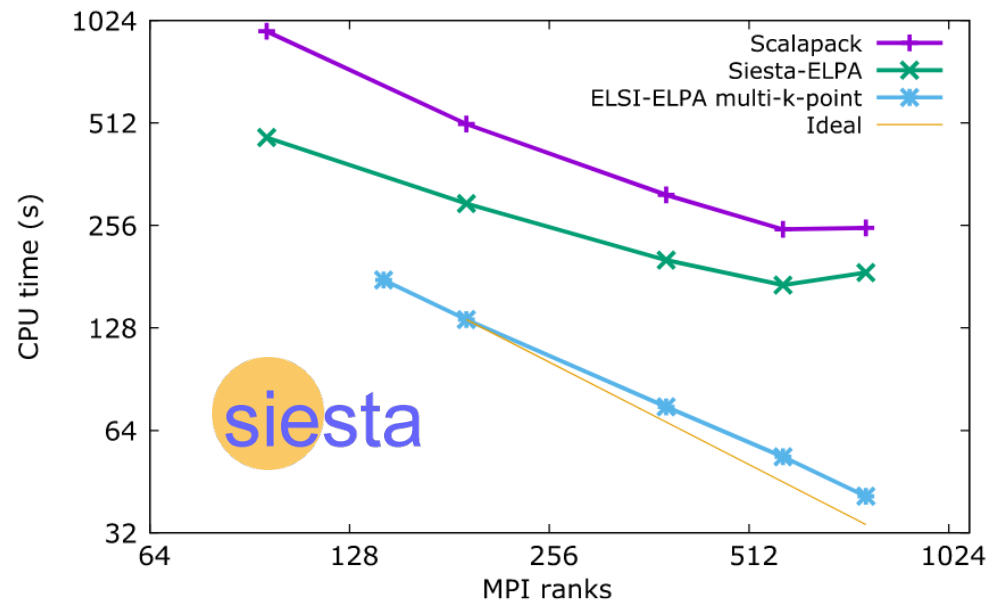
Modular code design



Electronic Structure Library



J. Chem. Phys.. 2020; 153(2). doi:10.1063/5.0012901



J. Chem. Phys. 152, 204108 (2020); doi: 10.1063/5.0005077



Documentation



Documentation



Good

Courses and tutorials

Documentation

Literature utilities

HPC

Cheatsheets

Teknowlogy

Communities

Utilities

Arch Linux

UU

Ground-State

Search

github-abinit
10.2.5 204 92

Variables

All Variables

Abinit

Basic

Bethe-Salpeter

Developers

DMFT

Electron-Phonon

Varff

File-handling

Geometry

Ground-State

GW

Internal

Parallelism

PAW

DFPT

Relaxation

VdW

Wannier90

Anaddb

aTDEP

Multibinit

Optic

Alm

External Parameters

kptrlatt

Mnemonics: K - PoinTs grid: Real space LATTice
Mentioned in topic(s): [topic_k-points](#)
Variable type: integer
Dimensions: (3,3)
Default value: 0
**The use of this variable forbids the use of:* [ngkpt](#)
Added in version: before_v9

Test list (click to open). Moderately used, [69/1245] in all abinit tests, [7/159] in abinit tutorials

This input variable is used only when [kptopt](#) is positive. It partially defines the k point grid. The other piece of information is contained in [shiftk](#). [kptrlatt](#) cannot be used together with [ngkpt](#).

The values [kptrlatt](#) (1:3,1), [kptrlatt](#) (1:3,2), [kptrlatt](#) (1:3,3) are the coordinates of three vectors in real space, expressed in the [%rprimd](#) coordinate system (reduced coordinates). They define a super-lattice in real space. The k point lattice is the reciprocal of this super-lattice, possibly shifted (see [shiftk](#)).

If neither [ngkpt](#) nor [kptrlatt](#) are defined, ABINIT will automatically generate a set of k point grids, and select the best combination of [kptrlatt](#) and [shiftk](#) that allows one to reach a sufficient value of [kptrlen](#). See this latter variable for a complete description of this procedure.

kptrlen

Mnemonics: K - PoinTs grid: Real space LENgth
Mentioned in topic(s): [topic_k-points](#)

Table of contents

[lockdownsampling](#)

[fockptmix](#)

[hyb_mixing](#)

[hyb_mixing_sr](#)

[hyb_range_dft](#)

[hyb_range_fock](#)

[iatsph](#)

[icoulomb](#)

[icutcoul](#)

[iprcel](#)

[iqpt](#)

[ivalence](#)

[ixcpositron](#)

[jellslab](#)

[kptbounds](#)

[kptrlatt](#)

[kptrlen](#)

[magcon_lambda](#)

[magconon](#)

[mixalch](#)

[natsph](#)

[natsph_extra](#)

[nbdbuf](#)

[nblock_jobpcg](#)

[ndivk](#)



Documentation

Good

Navigation: Courses and tutorials, Documentation, Literature utilities, HPC, Cheatsheets, Teknowlogy, Communities, Utilities, Arch Linux, UU

Ground-State

Search

github-abinit 10.2.4 204 92

Variables

All Variables

Abinit

Basic

Bethe-Salpeter

Developers

DMFT

Electron-Phonon

Varff

File-handling

Geometry

Ground-State

GW

Internal

Parallelism

PAW

DFPT

Relaxation

VdW

Wannier90

Anaddb

aTDEP

Multibinit

Optic

Aim

External Parameters

kptriatt

Mnemonics: K - PoinTs grid: Real space LATTice

Mentioned in topic(s): [topic_k-points](#)

Variable type: integer

Dimensions: (3,3)

Default value: 0

*The use of this variable forbids the use of: [ngkpt](#)

Added in version: before_v9

Test list (click to open). Moderately used, [69/1245] in all abinit tests, [7/159] in abinit tutorials

This input variable is used only when [kptopt](#) is positive. It partially defines the k point grid. The other piece of information is contained in [shiftk](#). [kptriatt](#) cannot be used together with [ngkpt](#).

The values [kptriatt](#) (1:3,1), [kptriatt](#) (1:3,2), [kptriatt](#) (1:3,3) are the coordinates of three vectors in real space, expressed in the [rprimd](#) coordinate system (reduced coordinates). They define a super-lattice in real space. The k point lattice is the reciprocal of this super-lattice, possibly shifted (see [shiftk](#)).

If neither [ngkpt](#) nor [kptriatt](#) are defined, ABINIT will automatically generate a set of k point grids, and select the best combination of [kptriatt](#) and [shiftk](#) that allows one to reach a sufficient value of [kptrlen](#). See this latter variable for a complete description of this procedure.

kptrlen

Mnemonics: K - PoinTs grid: Real space LENGTH

Mentioned in topic(s): [topic_k-points](#)

Table of contents

- [lockdownsampling](#)
- [fockoptmix](#)
- [hyb_mixing](#)
- [hyb_mixing_sr](#)
- [hyb_range_dft](#)
- [hyb_range_fock](#)
- [iatsph](#)
- [icoulomb](#)
- [icutcoul](#)
- [iprcel](#)
- [iqpt](#)
- [ivalence](#)
- [ixcpositron](#)
- [jellslab](#)
- [kptbounds](#)
- [kptriatt](#)
- [kptrlen](#)
- [magcon_lambda](#)
- [magconon](#)
- [mixalch](#)
- [natsph](#)
- [natsph_extra](#)
- [nbdubf](#)
- [nblock_jobpcg](#)
- [ndivk](#)

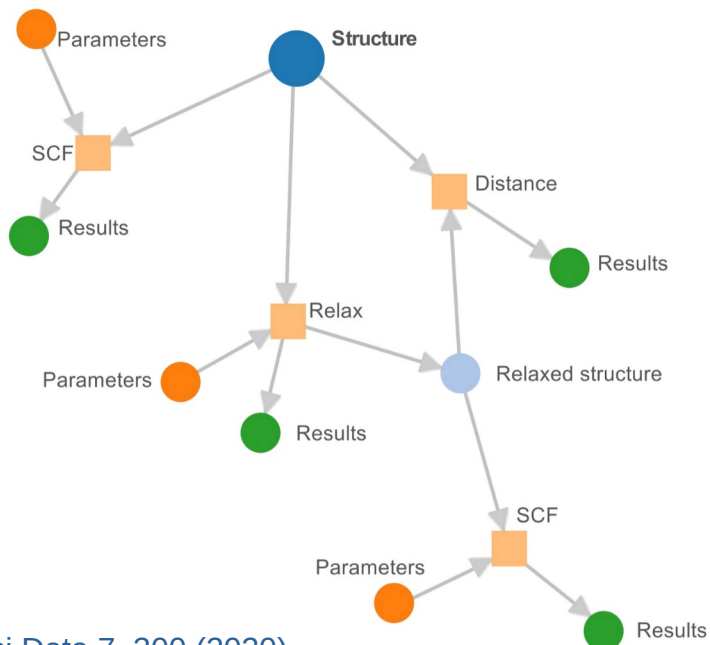
Bad

```
56 stop
57 endif
58
59 call POPBLASINFO( OUTFILE, nout, M, N, K, nb, NPROW, NPCOL, work, &
60               iam, NPROCS )
61 !
62 ! Define process grid.
63 !
64 call BLACS_GET( -1, 0, ICTXT )
65 call BLACS_GRIDINIT( ICTXT, 'Row-major', NPROW, NPCOL )
66 call BLACS_GRIDINFO( ICTXT, NPROW, NPCOL, my_row, my_col )
67 !
68 ! Go to bottom of process grid loop if this case doesn't use my
69 ! process.
70 !
71 if ( my_row >= NPROW .or. my_col >= NPCOL ) &
72   go to 20
73
74 MP = numroc( M, nb, my_row, 0, NPROW )
75 KP = numroc( K, nb, my_row, 0, NPROW )
76 KQ = numroc( K, nb, my_col, 0, NPCOL )
77 nq = numroc( N, nb, my_col, 0, NPCOL )
78 !
79 ! Initialize the array descriptor for the matrix A, B and C.
80 !
81 call descint( desca, M, K, nb, nb, 0, 0, ICTXT, MAX( 1, MP ), &
82             INFO )
83 call descint( descb, K, N, nb, nb, 0, 0, ICTXT, MAX( 1, KP ), &
84             INFO )
85 call descint( descc, M, N, nb, nb, 0, 0, ICTXT, MAX( 1, MP ), &
86             INFO )
87 !
88 ! Assign pointers into mem for SCALAPACK arrays, A is
89 ! allocated starting at position mem( 1 ).
90 !
91 IPA = 1
92 IPB = IPA + desca( LLD_ ) * KQ
93 IPC = IPB + descb( LLD_ ) * NQ
94 IPW = IPC + descc( LLD_ ) * NQ
95
96 WORKSZ = nb
```

FAIR data



F_{indable} A_{ccessible} I_{nteroperable} R_{eusable}



MATERIALSCLOUD



LEARN



WORK



DISCOVER



EXPLORE



ARCHIVE

Open-source Python
Infrastructure to

automate, manage, share and reproduce

- Complex computational workflows & data
- Designed for high-throughput workflows
- Interfaced with multiple material science codes
- Ensures full data provenance
- Allows for FAIR data storage



Future

GPU



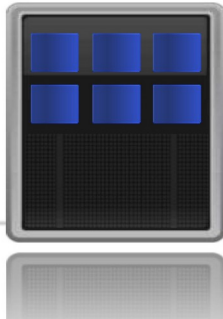
J. Chem. Theory Comput. 2023,
19, 6992–7006

GPU VS CPU

GPU 10X PERFORMANCE & 5X ENERGY EFFICIENCY

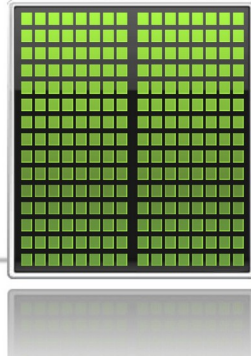
CPU

Few Cores
Optimized for
Serial Tasks



GPU Accelerator

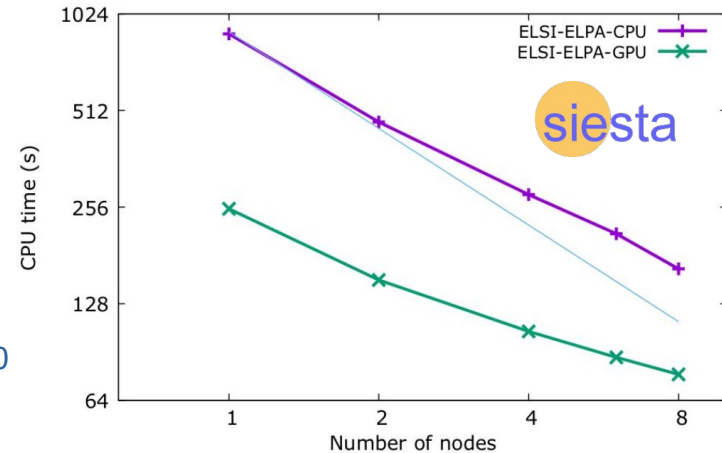
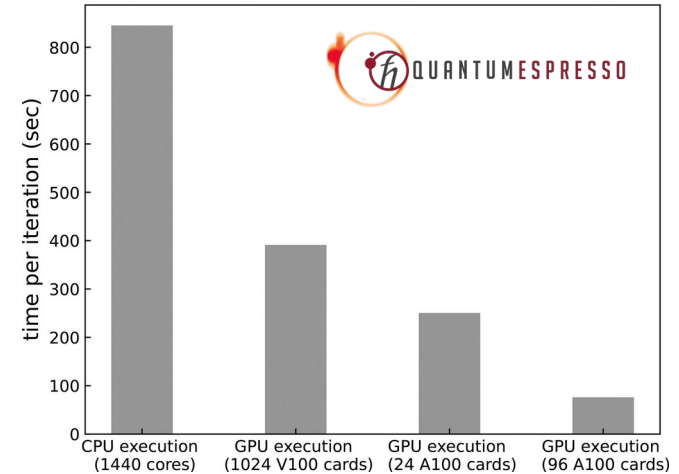
1000's of Cores
Optimized for
Parallel Tasks



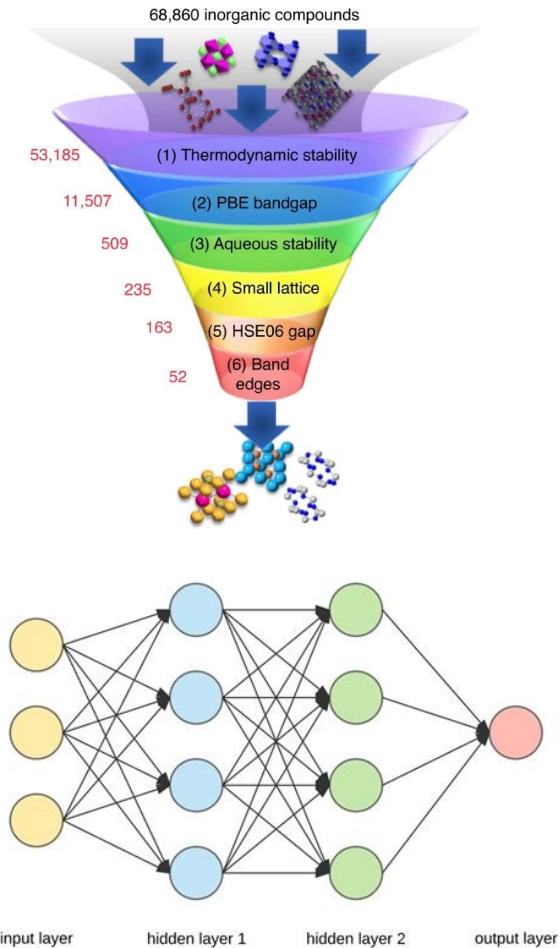
Horizon MaX report 2020

a.h.kole@uu.nl

PWSCF - Chromium Iodide orthorhombic supercell
(1152 atoms, 7776 electrons, NCPP+CC:240/60Ry)



Workflows for high throughput



The Materials Project

Home / Apps / Materials Explorer

Materials Explorer

Search for materials information by chemistry, composition, or property.

Materials e.g. Li-Fe or Li,Fe or Li3Fe or mp-19017

Only Elements At Least Elements Formula

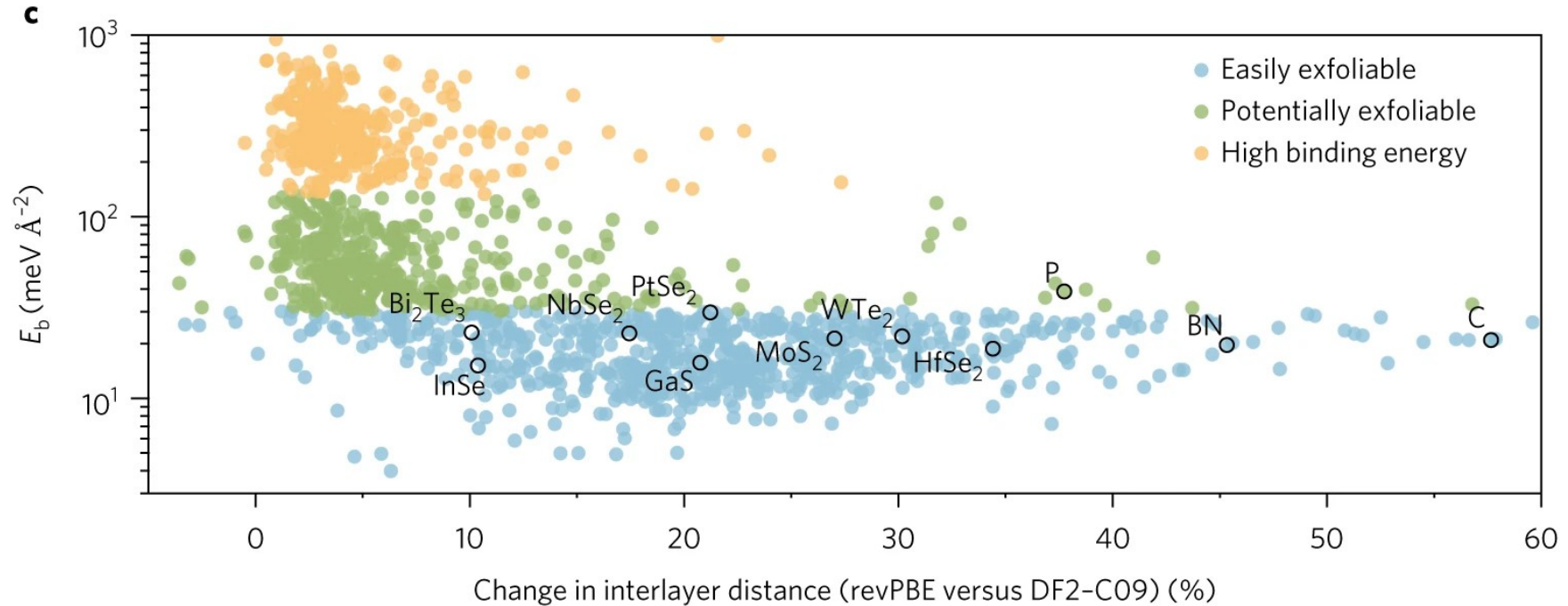
Select elements to search for materials with only these elements

Filters Reset

All 153,235 materials
Showing 1-15

Material ID	Formula	Crystal System	Space Group Symbol	Sites	E
-------------	---------	----------------	--------------------	-------	---

Workflows for high throughput

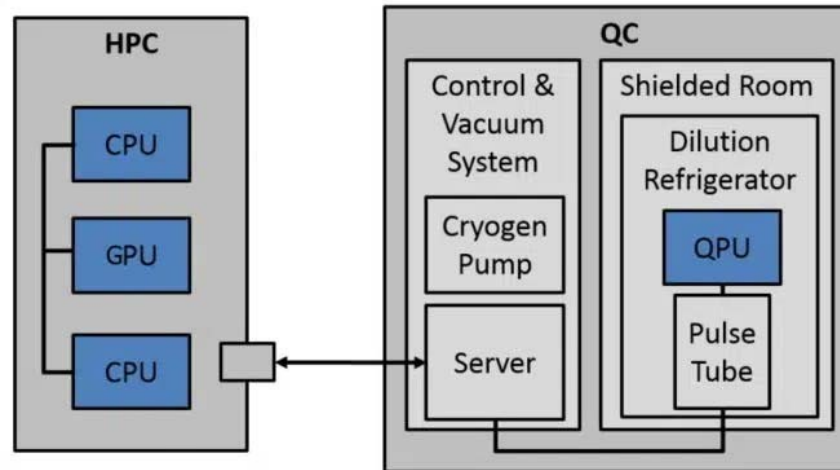


Nature Nanotechnology volume 13, pages 246–252 (2018)

Quantum Computing



- Perfect fit for solving quantum problem
- Current applicability still limited => hybrid algorithms
- Using regular (CPU+GPU) and quantum (QPU) hardware in tandem

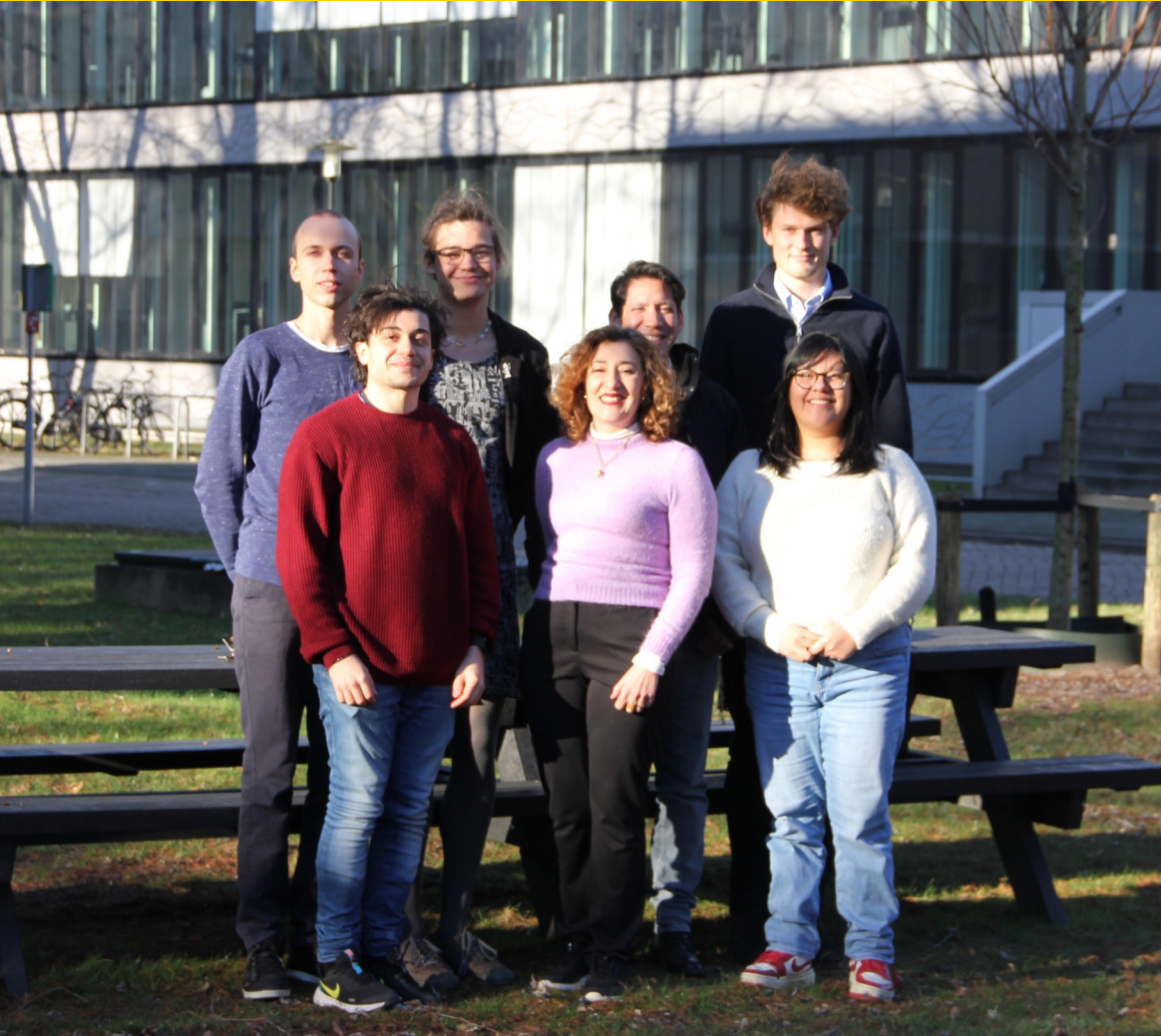


Summary

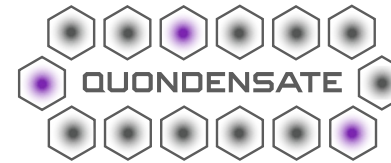


- Quantum Materials exhibit quantum behaviour at macroscopic scales
- Quantum description using first-principles simulations
- Compute properties in a parameter-free way to aid materials design
- Recent extension to include superconductivity
- Open-source codes that scale well on High Performance Computing infrastructure
- Good practices to help the community

Quantum Materials by Design 2024 Team



Funding



European
Innovation
Council

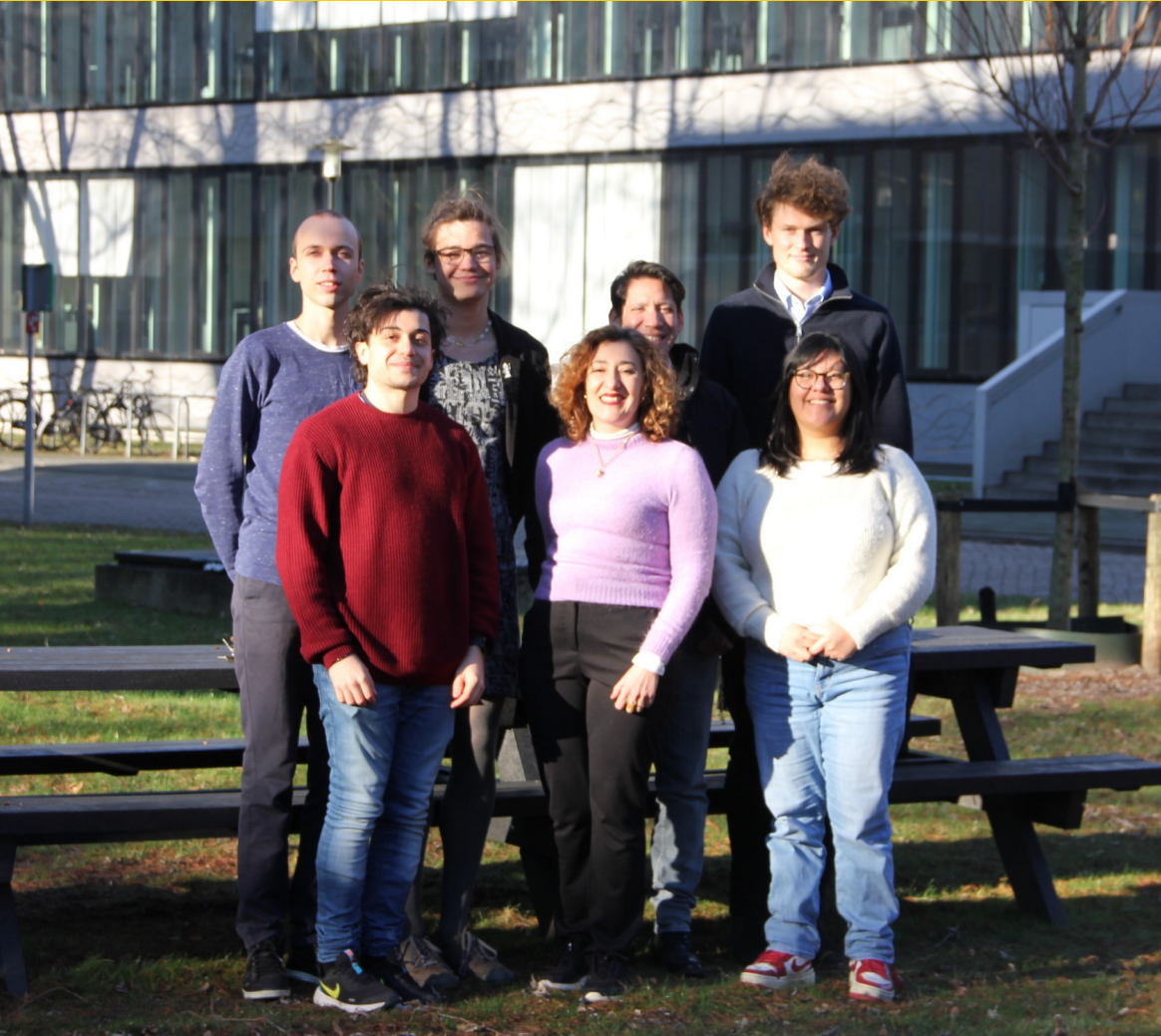


PARTNERSHIP FOR ADVANCED
COMPUTING IN EUROPE

Codes



Quantum Materials by Design 2024 Team



Projects on Snellius

- NWO-2022.012
- EINF-6220
- NWO-2024.012



Publications

Sohier 2D Mat 10(2), 025006, 2023
Melo npj CM 9(1), 147, 2023
Zanolli A&A 675, L9, 2023
Moes NL 24(17), 5110-5116, 2024
Reho PRB 110 (3), 035118, 2024
Reho PRB 110 (13), 134505, 2024
Reho arxiv 2024
Cuperus submitted 2024
Vonk submitted 2024
Reho in preparation 2025

Student theses

Wittemeier 2023
Tops 2024
Dubbink 2024

14 keynote and invited presentations at international conferences

Questions?