SIMLAB QUANTUM MATERIALS

A model for domain-specific sustainable support and development

July 8, 2019 | Edoardo Di Napoli |

OUTLINE

Quantum Materials in a nutshell

Perspectives and trends

Challenges and solutions

The SL Quantum Materials

OUTLINE

Quantum Materials in a nutshell

Perspectives and trends

Challenges and solutions

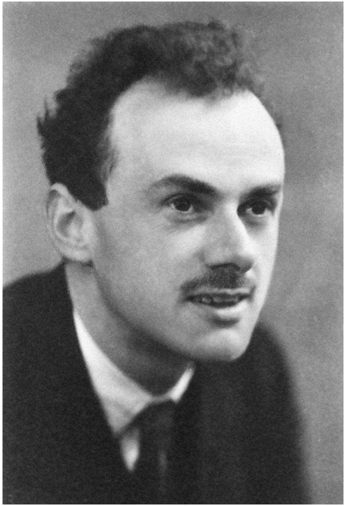
The SL Quantum Materials

JÜLICH Forschungszentrum

IT STARTED WITH E. SCHRÖDINGER . . .

$$i\hbar\frac{\partial\Psi}{\partial t} = H\Psi$$

. . . IT ENDED WITH P.A.M. DIRAC.



“The underlying **physical laws** *necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus* **completely known***, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble. It therefore becomes desirable that* **approximate practical methods** *of applying quantum mechanics should be developed, which can lead to an explanation of the main features of complex atomic systems* **without too much computation***”.*

JÜLICH Forschungszentrum

APPROXIMATE SOLUTIONS

A plethora of methods

Definition

Ab initio is a Latin term meaning “from the beginning” and is derived from the Latin ab (“from”) + initio, ablative singular of initium (“beginning”).

Ab initio molecular orbitals methods:

- Hartree-Fock – # ops scales as $\sim N^4$
- $\left.\begin{array}{l}\text{Møller-Plesset}\\ \text{Configuration Interaction (CI)}\\ \text{Coupled Clusters (CC)}\\ \text{etc.}\end{array}\right\}$ – # ops scales as $N^4 \div N^7$

Ab initio electron density methods:

- Density Functional Theory (DFT) – # ops scales as $N \ln N \div N^3$
- Car-Parrinello Molecular Dynamics (MD)

OUTLINE

Quantum Materials in a nutshell

Perspectives and trends

Challenges and solutions

The SL Quantum Materials

TRENDS IN ACADEMIA AND INDUSTRY

Private sector:

1. **Target (an example)**: the restructuring of the energy system by investigating materials for energy conversion processes and storage technologies.
2. **Method**: Using HTC for a large number of small scale simulations aimed at low impact materials screening
3. **Path**: Stable and verified simulation software (license

   preferred).
4. **Time frame**: 3 to 6 months

Public sector across topics and disciplines

1. **Target**: Fundamental research in method and functionality development
2. **Method**: Using large scale computations to simulate real world materials
3. **Path**: Code implementation without concern for abstractions and efficiency
4. **Time frame**: 1 to 3 years

TRENDS IN THE SUPERCOMPUTING INDUSTRY

Summit (ORNL)

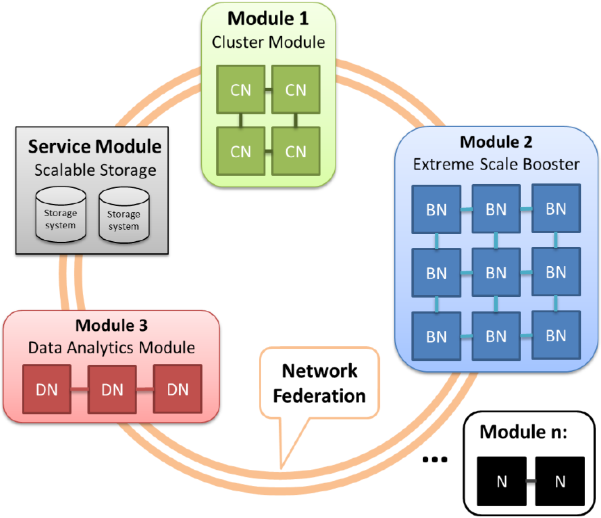
- Manufacturer: IBM
- Processor: IBM POWER9 (2/node) 3.07GHz
- GPUs: 27,648 NVIDIA Volta V100s (6/node)
- Cores: 2,414,592 – Nodes: 4,608
- Memory/node: 512GB DDR4 + 96GB HBM2
- Interconnect: Mellanox EDR Infiniband
- Peak performance: 200 PFlop/s

SuperMUC-NG (LRZ)

- Manufacturer: Lenovo
- Processor: Xeon Skylake Platinum 3.1GHz
- GPUs: –
- Cores: 311,040 – Nodes: 6,480
- Memory/node: 96 GB
- Interconnect: Intel Omni-Path
- Peak performance: 27 PFlop/s

TRENDS IN THE SUPERCOMPUTING INDUSTRY

Modular Supercomputer



TRENDS IN THE SUPERCOMPUTING INDUSTRY

Modular Supercomputer

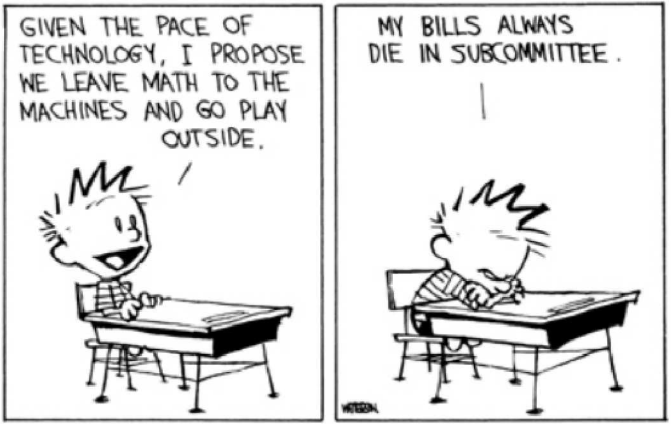
JUWELS Module 1 (FZJ)

- Manufacturer: Bull
- Processor: Xeon Skylake Platinum 2.7GHz
- GPUs: –
- Cores: 122,448 – Nodes: 2,567
- Memory/node: 96GB DDR4
- Interconnect: Mellanox EDR Infiniband
- Peak performance: 9 PFlop/s

JUWELS Module 2 (FZJ)

- Manufacturer: ??
- Processor: Server Class
- GPUs: 4GPUs/node
- Cores: ??
- Memory/node: ?? GB
- Interconnect: at least 800Gbit/s
- Peak performance: $\sim$ 100 PFlop/s

YOU MAY FEEL LIKE CALVIN . . .



OUTLINE

Quantum Materials in a nutshell

Perspectives and trends

Challenges and solutions

The SL Quantum Materials

HIGH-PERFORMANCE COMPUTATIONS

Observations:

- Numerical algorithms are ubiquitous in *Condensed Matter Physics*
- Numerical libraries are used as **black boxes**.
- Domain-specific **knowledge** does not influence algorithm choice.
- Rigid legacy codes are hard to **modernize**.

Goal

Design and optimize linear algebra algorithms in order to:

- exploit available **knowledge**.
- increase the **parallelism** of complex tasks.
- facilitate **performance portability**

HPC TRANSFER

Issues and workflows

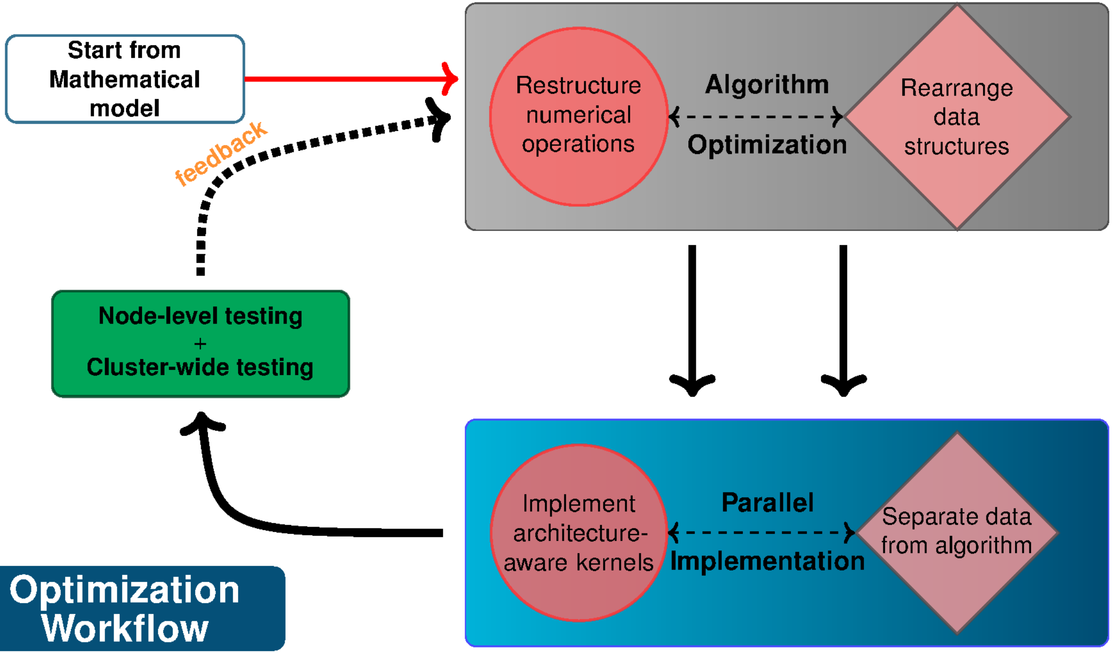
Legacy codes

- Conceived with a rigid abstraction model
- Developed by application scientists in isolation
- Carry the “curse of early optimization”
- Non-modular, often written in FORTRAN
- Many, many functionalities

Refactoring and modernization

- Code modernization
- Algorithm optimization
- Performance portability
- Code refactoring
- Inclusion of domain-specific knowledge

Optimization Workflow



THE FLAPW MATHEMATICAL MODEL

Generation of H and S

Set of basis functions

$$\varphi_t(\mathbf{r}) = \begin{cases} \sum_{l=0}^{l=l_{\max}} \sum_{m=-l}^{m=+l} \left[A_{(l,m),a,t} u_{l,a}(r) + B_{(l,m),a,t} \dot{u}_{l,a}(r) \right] Y_{l,m}(\hat{\mathbf{r}}_a) & a^{\text{th}}\ MT \\ \frac{1}{\sqrt{\Omega}} \exp(i\mathbf{K}_t \cdot \mathbf{r}) & INT \end{cases} \quad (1)$$

Kohn-Sham Hamiltonian and Overlap matrices

$$(H)_{t',t} = \sum_a \iint \varphi^*_{t'}(\mathbf{r}) \hat{H}_{\mathrm{KS}} \varphi_t(\mathbf{r}) \mathrm{d}\mathbf{r}, \quad (S)_{t',t} = \sum_a \iint \varphi^*_{t'}(\mathbf{r}) \varphi_t(\mathbf{r}) \mathrm{d}\mathbf{r}. \quad (2)$$

THE FLAPW MATHEMATICAL MODEL

Generation of H and S

The Hamiltonian matrix

$$
\begin{aligned}
(H)_{t',t} = \sum_{a} \sum_{L',L} &\left(A^{*}_{L',a,t'} \, T^{[AA]}_{L',L;a} A_{L,a,t} \right) + \left(A^{*}_{L',a,t'} \, T^{[AB]}_{L',L;a} B_{L,a,t} \right) \\
&+ \left(B^{*}_{L',a,t'} \, T^{[BA]}_{L',L;a} A_{L,a,t} \right) + \left(B^{*}_{L',a,t'} \, T^{[BB]}_{L',L;a} B_{L,a,t} \right) .
\end{aligned} \tag{3}
$$

The new matrices $T^{[\ldots]}_{L',L;a} \in \mathbb{C}^{N_L \times N_L}$ are dense as well and their computation involves multiple integrals between the basis functions and the non-spherical part of the potential V_{eff}.

The Overlap matrix

$$
(S)_{t',t} = \sum_{a} \sum_{L=(l,m)} A^{*}_{L,a,t'} A_{L,a,t} + B^{*}_{L,a,t'} B_{L,a,t} \left\| \dot{u}_{l,a} \right\|^{2} \tag{4}
$$

HAMILTONIAN AND OVERLAP MATRIX GENERATION.

Matrix visualization

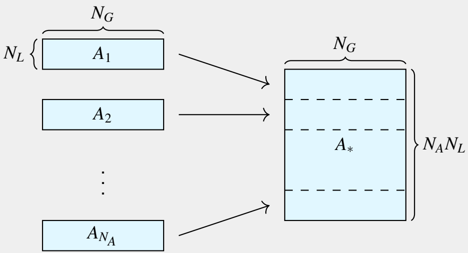
$$H = \underbrace{\sum_{a=1}^{N_A} A_a^H T_a^{[AA]} A_a}_{H_{AA}} + \underbrace{A_a^H T_a^{[AB]} B_a + B_a^H T_a^{[BA]} A_a + B_a^H T_a^{[BB]} B_a}_{H_{AB+BA+BB}}$$

$$S = \underbrace{\sum_{a=1}^{N_A} A_a^H A_a}_{S_{AA}} + \underbrace{\sum_{a=1}^{N_A} B_a^H \dot{U}_a^H \dot{U}_a B_a}_{S_{BB}}$$

A_a and $B_a \in \mathbb{C}^{N_L \times N_G}$ while $T_a^{\cdots} \in \mathbb{C}^{N_L \times N_L}$ and Hermitian.
$N_L = (l_{\max}+1)^2 \leq 121$, $N_G = 1,000 \div 50,000$, and $N_A =$ number of atoms.

HAMILTONIAN AND OVERLAP MATRIX GENERATION

Data layout



1. $S \mathrel{+}= A_*^H A_*$ (`zherk`: $4N_A N_L N_G^2$ FLOPs)
2. $B_* =: \dot{U}_* B_*$ ($2N_A N_L N_G$ FLOPs)
3. $S \mathrel{+}= B_*^H B_*$ (`zherk`: $4N_A N_L N_G^2$ FLOPs)

JÜLICH Forschungszentrum

PORTED TO HYBRID PLATFORMS (CPU+GPU)

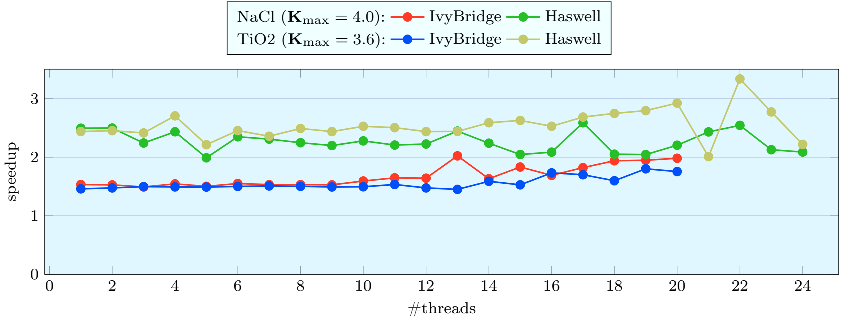


Figure 1: Speedup of our algorithm over FLEUR with $k_{max} = 4$ and increasing parallelism

EDN, E. Peise, M. Hrywniak, and P. Bientinesi. Comp. Phys. Comm. 211 (2017), pp. 61-72, [arXiv:1602.06589].

Davor Davidovìc, D. Fabregat-Traver, M. Höhnerbach, EDN. Concurrency Computat. Pract Exper. 30(24), e4905 (2018) doi: 10.1002/cpe.4905, [arXiv:1712.07206]

PORTED TO HYBRID PLATFORMS (CPU+GPU)

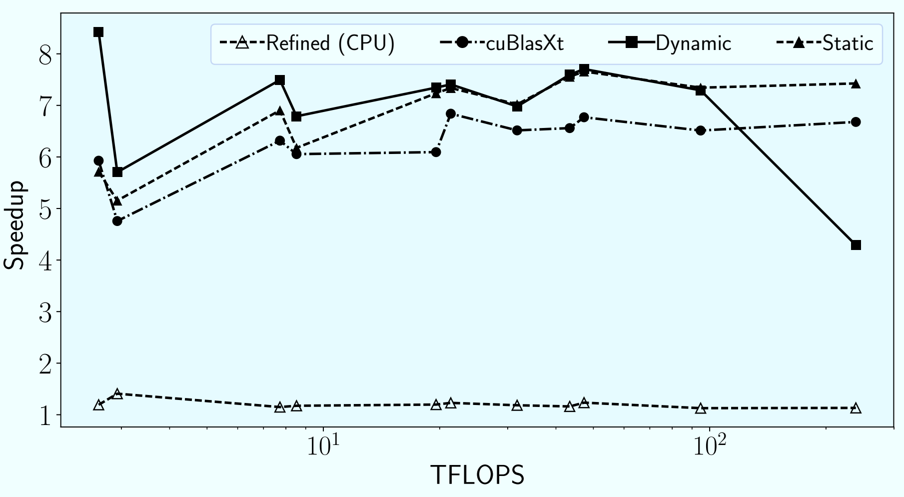


Figure: Speedup on RWTH-GPU for all implementations, relative to original code.

EDN, E. Pelse, M. Hrywniak, and P. Bientinesi. Comp. Phys. Comm. 211 (2017), pp. 61-72, [arXiv:1602.06589].

Davor Davidoviò, D. Fabregat-Traver, M. Höhnerbach, EDN, Concurrency Computat. Pract Exper. 30(24), e4905 (2018) doi: 10.1002/cpe.4905, [arXiv:1712.07206].

PARALLEL NUMERICAL INTEGRATION IN TUFRG

The mathematical equations

$$\dot{\mathbf{P}}(\mathbf{l}) = \mathbf{V}^P(\mathbf{l})\,\dot{\mathcal{X}}^{\mathrm{pp}}(\mathbf{l})\,\mathbf{V}^P(\mathbf{l})\,, \tag{5}$$

$$\dot{\mathbf{C}}(\mathbf{l}) = -\mathbf{V}^C(\mathbf{l})\,\dot{\mathcal{X}}^{\mathrm{ph}}(\mathbf{l})\,\mathbf{V}^C(\mathbf{l})\,, \tag{6}$$

$$\dot{\mathbf{D}}(\mathbf{l}) = 2\mathbf{V}^D(\mathbf{l})\,\dot{\mathcal{X}}^{\mathrm{ph}}(\mathbf{l})\,\mathbf{V}^D(\mathbf{l}) - \mathbf{V}^C(\mathbf{l})\,\dot{\mathcal{X}}^{\mathrm{ph}}(\mathbf{l})\,\mathbf{V}^D(\mathbf{l}) - \mathbf{V}^D(\mathbf{l})\,\dot{\mathcal{X}}^{\mathrm{ph}}(\mathbf{l})\,\mathbf{V}^C(\mathbf{l})\,, \tag{7}$$

where

$$\mathcal{X}^{\mathrm{pp}}_{m,n}(\mathbf{l}) = \int d\mathbf{p}\left[\int dp_0\, G\left(p_0, \frac{\mathbf{l}}{2} + \mathbf{p}\right) G\left(-p_0, \frac{\mathbf{l}}{2} - \mathbf{p}\right)\right] f_m(\mathbf{p}) f_n(\mathbf{p})\,, \tag{8}$$

$$\mathcal{X}^{\mathrm{ph}}_{m,n}(\mathbf{l}) = \int d\mathbf{p}\left[\int dp_0\, G\left(p_0, \mathbf{p} + \frac{\mathbf{l}}{2}\right) G\left(p_0, \mathbf{p} - \frac{\mathbf{l}}{2}\right)\right] f_m(\mathbf{p}) f_n(\mathbf{p})\,. \tag{9}$$

PARALLEL NUMERICAL INTEGRATION IN TUFRG

Where the computing time goes?

Assemble interaction

$P, C, D \to V^P$

~ 20**% CPU Time**

$\longrightarrow$

Perform 2D integration

$\dot{\chi}^{\mathrm{pp}}_{m,n}(\mathbf{l}) \quad \forall\, m,n,\mathbf{l}$

~ 80**% CPU Time**

$\downarrow$

Matrix multiplication

$\sum_{p,q} V^P_{m,p}(\mathbf{l})\, \dot{\chi}^{\mathrm{pp}}_{p,q}(\mathbf{l})\, V^P_{q,n}(\mathbf{l})$

< 1% CPU Time

$\longleftarrow$

Iterate ODE for P, C, D

$\frac{d}{d\Omega} P_{m,n}(\mathbf{l})$

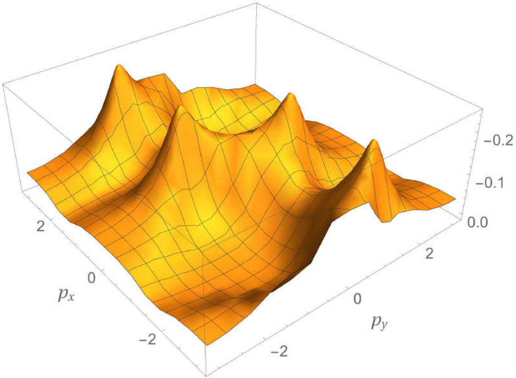
< 1% CPU Time

$\uparrow$

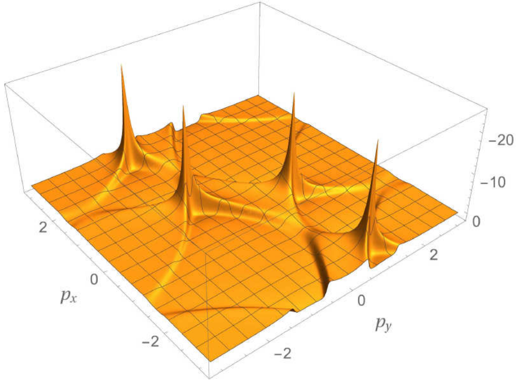
PARALLEL NUMERICAL INTEGRATION IN TUFRG

Exploring knowledge inclusion

$\Omega = 1.0$



$\Omega = 0.1$



JÜLICH Forschungszentrum

PARALLEL NUMERICAL INTEGRATION IN TUFRG

Where does the accuracy go?

Error definition

$$\mathsf{ERR}[\phi] = |Q_{n_1}\phi - Q_{n_2}\phi|,$$

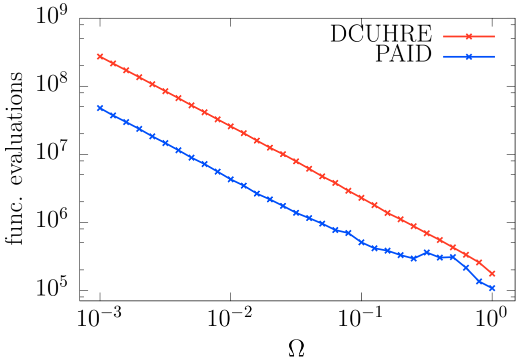
where with $Q_n\phi = Q(\phi, \mathcal{D}, n)$ we indicate the computation of the integral $\Phi = \int_{\mathcal{D}}\phi$ over the domain $\mathcal{D}$ through numerical quadrature with n integration points

$$\begin{aligned}\mathsf{ERR}[\dot{P}_{m,n}] &= \left|\sum_{p,q}\left[V^P_{m,p}\left(Q_N\,\dot{\chi}^{\mathrm{pp}}_{p,q}\right)V^P_{q,n} - V^P_{m,p}\left(Q_{2N}\,\dot{\chi}^{\mathrm{pp}}_{p,q}\right)V^P_{q,n}\right]\right| \\ &\leq \left\|V^P_{m,:}\right\|_\infty \left\|V^P_{:,n}\right\|_\infty \sum_{p,q}\left|Q_N\,\dot{\chi}^{\mathrm{pp}}_{p,q} - Q_{2N}\,\dot{\chi}^{\mathrm{pp}}_{p,q}\right|,\end{aligned} \tag{10}$$

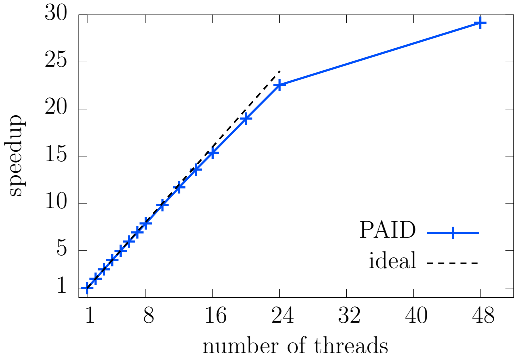
PARALLEL NUMERICAL INTEGRATION IN TUFRG

Computing less is computing better

Function evaluations



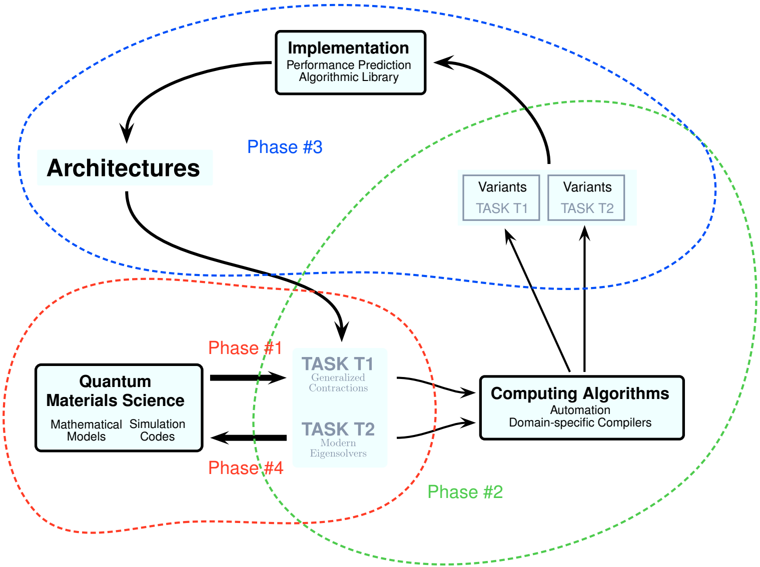
Node-level performance



J. Lichtenstein, J. Winkelmann, D. Sanchez de la Pena, Toni Vidovìc, EDN. Lecture Notes in Computer Science, High-Performance Scientific Computing 10164 (2016), pp. 170-184,

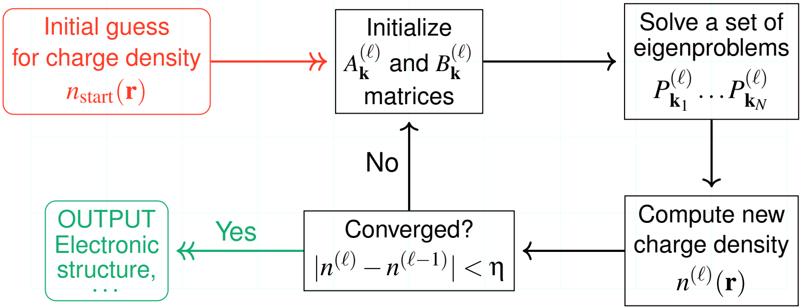
[arXiv:1610.09991]

A DEEPEST WORKFLOW



DFT SELF-CONSISTENT FIELD

General framework



THE CASE OF THE FLAPW METHOD

Observations:

1. every $P_{\mathbf{k}}^{(\ell)} : A_{\mathbf{k}}^{(\ell)} x = B_{\mathbf{k}}^{(\ell)} \lambda x$ is a generalized eigenvalue problem;
2. A and B are hermitian (B is positive definite);
3. required: lower $2 \div 20$ % of eigenpairs;
4. eigenvectors of problems of same **k** are seemingly uncorrelated across iterations i
5. k-vector index: $\mathbf{k} = 1 : 10 \div 100$;
6. iteration cycle index: $\ell = 1 : 20 \div 50$.

EIGENSOLVER ALGORITHM: SUBSPACE ITERATION

Two distinct optmization opportunities:

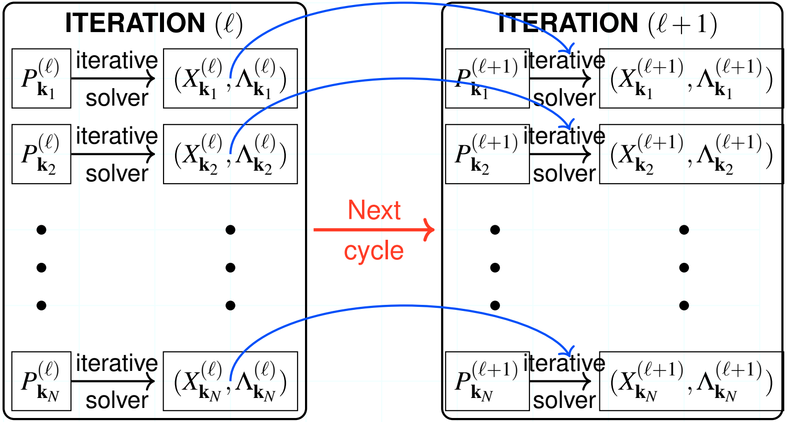
Knowledge exploitation:
Solution of previous SCF cycle can be used to “precondition” the solver at the next iteration

Algorithm optimization:
Polynomial degree can be pre-computed in order to minimize Mat-Vec multiplications

JÜLICH Forschungszentrum

EXPLOITING KNOWLEDGE

Preconditioning the eigensolver



$X \equiv \{x_1, \ldots, x_n\}$ $\quad$ $\Lambda \equiv \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$

A KNOWLEDGE-INCLUSIVE OPTIMIZED EIGENSOLVER

ChASE
CHEBYSHEV ACCELERATED SUBSPACE EIGENSOLVER

- License: open source — BSD 2.0
- GitHub: `https://github.com/SimLabQuantumMaterials/ChASE`
- Docs: `https://simlabquantummaterials.github.io/ChASE/index.html`
- Reference: `https://doi.org/10.1145/3313828`

Highlights

- Sequences of correlated eigenvalue problems
- Modern C++ interface: easy-to-integrate, performance portable
- Excellent strong- and weak-scale performance

JÜLICH Forschungszentrum

OUTLINE

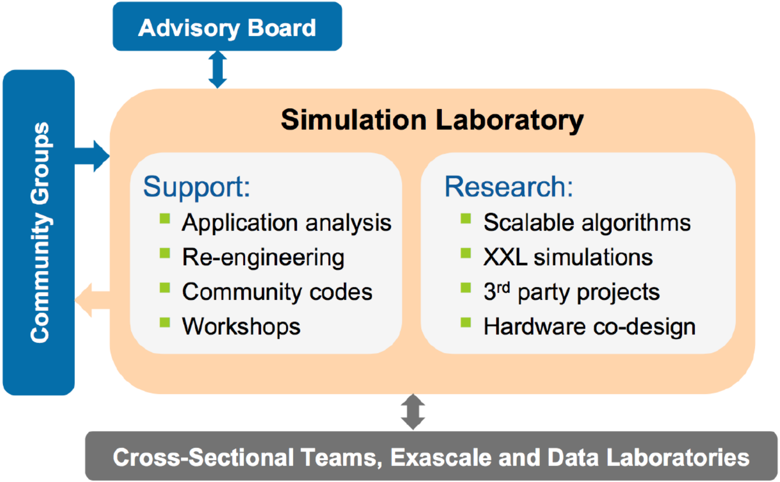
Quantum Materials in a nutshell

Perspectives and trends

Challenges and solutions

The SL Quantum Materials

SIMULATION LABORATORY AS HPC ENABLER



SIMLAB QUANTUM MATERIALS (SLQM)

SLQM area of activities

- Method development, algorithmic design, and code modernization
- HPC Knowledge transfer
- Programming models, Performance portability, Hybrid architectures, etc..

Mission

The Simulation Laboratory Quantum Materials (SLQM) provides expertise in the field of quantum-based simulations in Materials Science with a special focus on high-performance computing. SLQM acts as a high-level support structure in dedicated projects and hosts research projects dealing with fundamental aspects of code development, algorithmic optimization, and performance improvement. The Lab acts as an enabler of large scale simulations on current HPC platforms as well as on future architectures by targeting domain-specific co-design processes.

THE SLQM TEAM

- Edoardo Di Napoli – Senior researcher
- Paul Baumeister – Senior researcher
- Daniel Rohe – Optimization and support
- Xinzhe Wu – Postdoctoral researcher
- Sebastian Achilles – Senior PhD
- Miriam Hinzen – Senior PhD
- Jonas Dedden – Master

THANK YOU

QUANTUMATERIALS

For more information

e.di.napoli@fz-juelich.de

http://www.fz-juelich.de/ias/jsc/slqm