

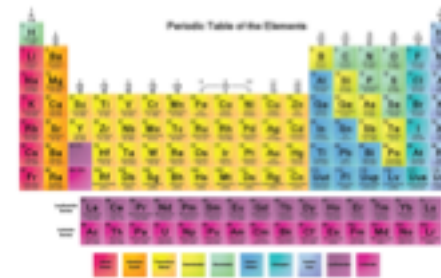
Towards Efficient Linear Scaling Green Function DFT Calculations

Restructuring KKRnano for modern HPC systems

2017 Oct 19 | Paul F. Baumeister¹

¹ Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany

Motivation



- Density Functional Theory (DFT) is a workhorse in materials science
- Relevant eigenstates of $H = T + V$ extend over the entire system
- Need to find one eigenstate per electron (pair) plus some unoccupied
- Eigenstates need to be kept orthogonal $\rightarrow$ scaling N^3_{atoms} unavoidable

Real systems feature broken symmetries
so we need to calculate for many atoms

- Green Function based DFT instead of wave functions $\rightarrow N^2_{\text{atoms}}$
- Distance-truncation of the Green function possible $\rightarrow$ linear in N_{atoms}

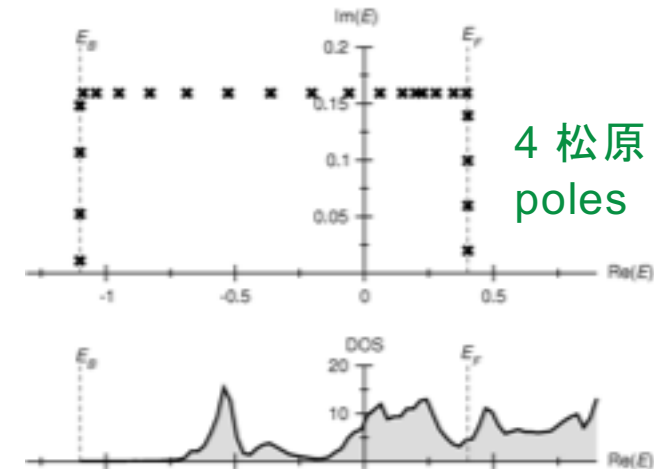
KKRnano

- Methodology based on Korringa-Kohn-Rostoker multiple scattering formalism
- Energy contour integration in the complex plane
- Iterative inversion of a “Hamiltonian” matrix

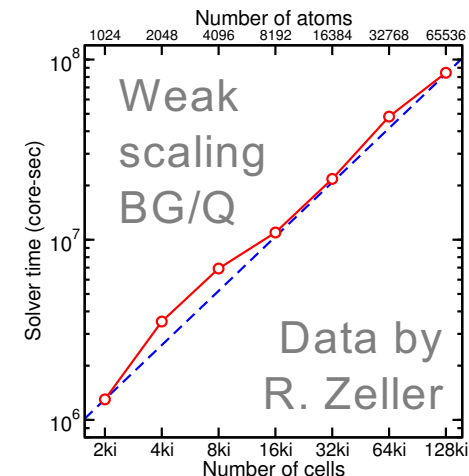
$$(E - H) \times G(\mathbf{r}, \mathbf{r}') == 1$$

- Multiple right hand sides treated in $G \rightarrow$ Matrix-matrix-multiplication instead of matrix-vector
- Total energy and atomic forces, non-collinear magnetism, magnetic fields
- MPI and OpenMP parallel Fortran90 code
- code generation (replacing C++ templates)

$T = 25000$ Kelvin



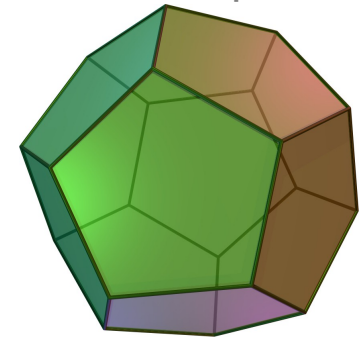
Plot by A. Thiess



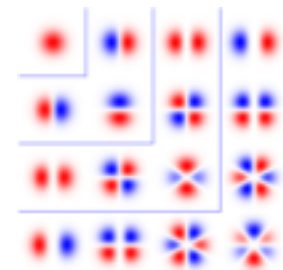
KKRnano - Representation

- Full potential all electron method
- Exponential radial grid up the innermost plane
- Space tessellation by Voronoi polyhedra around atom cores
- Radial grid of panels with equidistant mesh
- No FFT required
- Solves electrostatics via multipole expansion
- Most important convergence parameter is $\ell_{\max}$

Pic: Wikipedia



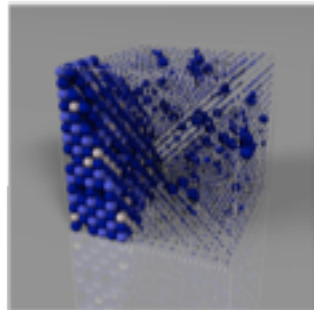
$\ell_{\max} = 3$
s, p, d, f



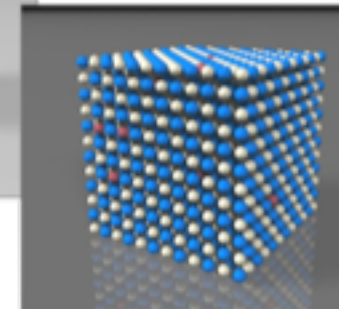
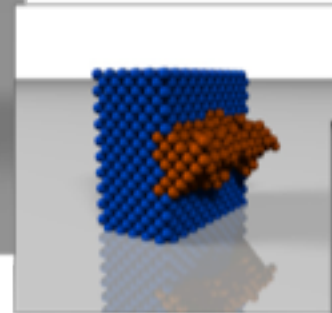
KKRnano – Application to Materials

- Disorder
- Impurities
- Small relaxations
- Grain boundaries
- Metallic glasses
- Magnetic textures

Phase-Change Materials
GeSbTe disorder & localization



Magnetic Metallic Compounds
 $\text{Fe}_x\text{Co}_{1-x}$ magnetic exchange
 Ni in Pd induced moments
 Cu in Ni Konbu-phase

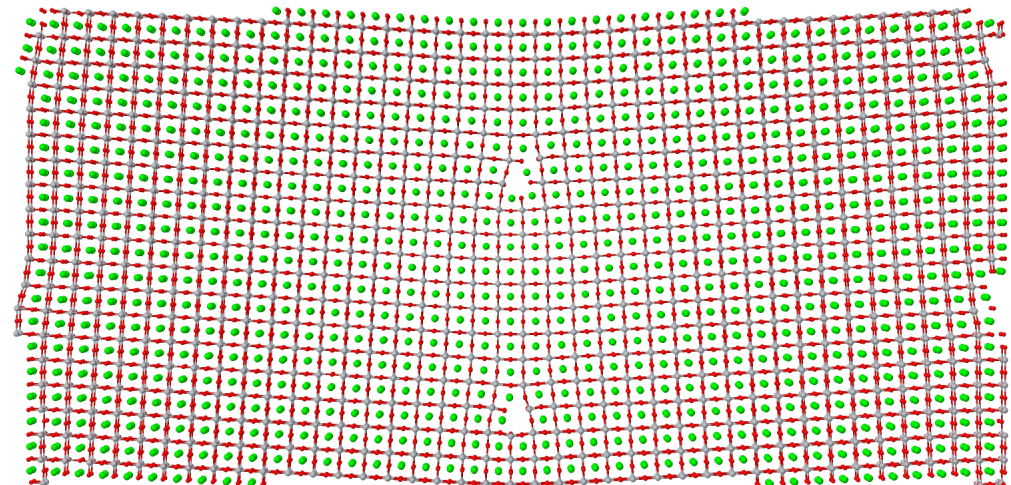
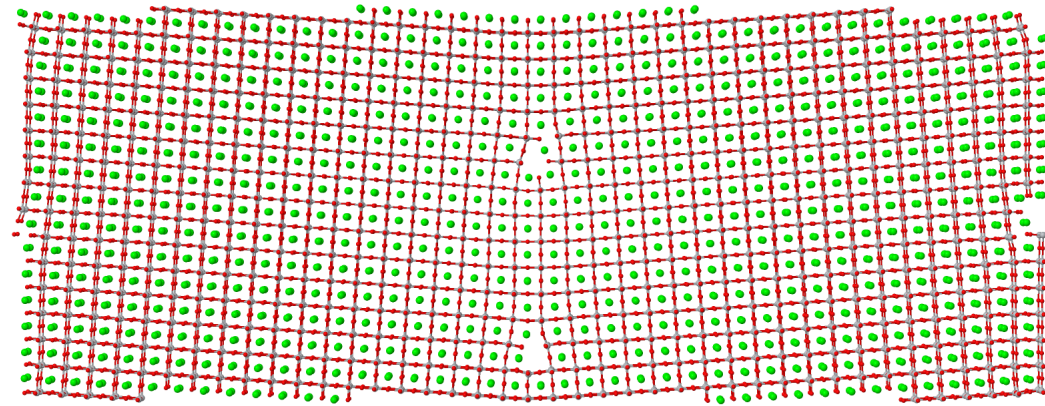


Dilute Magnetic Semiconductors
 MgO:N magnetic exchange
 GaN:Gd colossal moments

➔ Any sort of broken symmetry

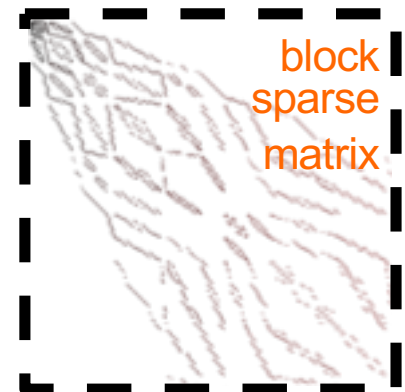
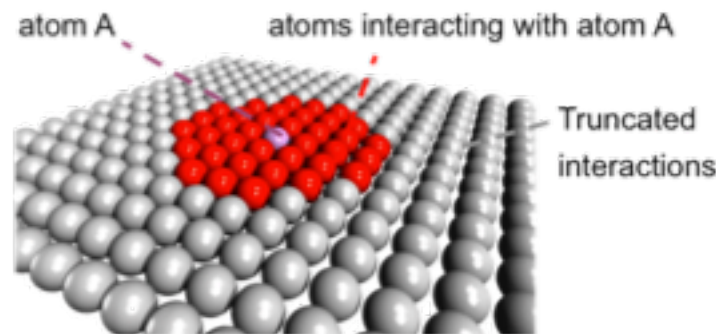
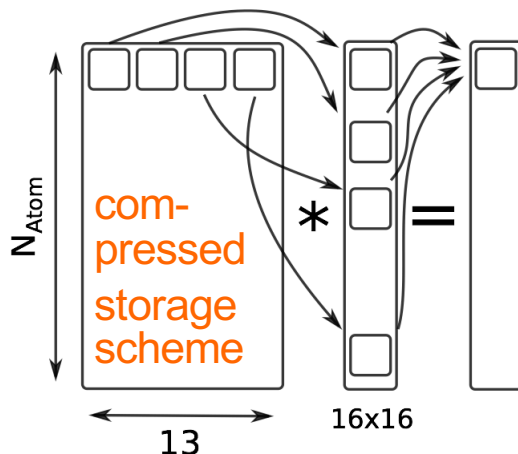
KKRnano – Application – Grain boundaries

- Small angle grain boundaries in SrTiO_3
- Two different setups,
 $7.6^\circ \rightarrow 2950$ atoms
 $6.0^\circ \rightarrow 3750$ atoms
- Placement of empty cells necessary
- Lloyd's formula needed for the correct E_{Fermi}



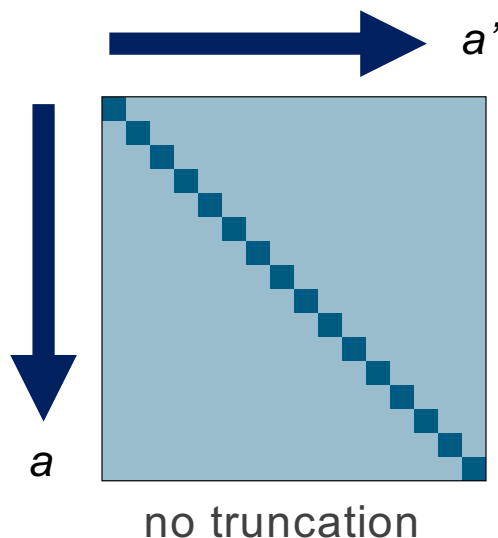
KKRnano – Linear Scaling by Truncation

- Green function elements are truncated when $|R_i - R_j| > R_{\text{cut}}$
- Fast convergence for insulators ($\rightarrow$ Density matrix methods)
- Kohn-nearsightedness principle allows also metals
- Convergence of relevant quantities for $N_{\text{trunc}} \cong 1000$ atoms
- Cost scales as $N_{\text{atoms}} \times N_{\text{trunc}} \times \text{Slowdown}$



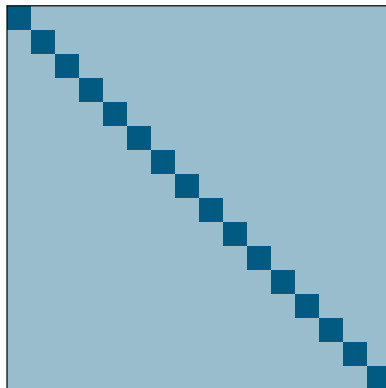
Truncation of Green function elements

- Truncation of interactions of very far Voronoi cells
- Test system `long_cell_with_truncation` consists of a linear periodic chain of 16 atoms.

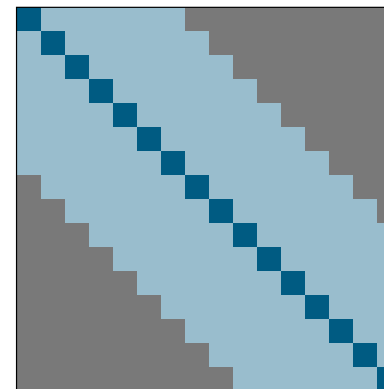


Truncation of Green function elements

- Truncation of interactions of very far Voronoi cells
- Test system `long_cell_with_truncation` consists of a linear periodic chain of 16 atoms.
- Truncation switched on: `cutoff_radius = 6.9`



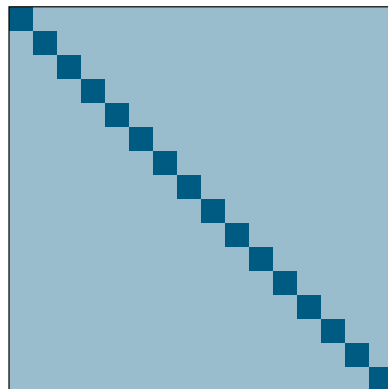
no truncation



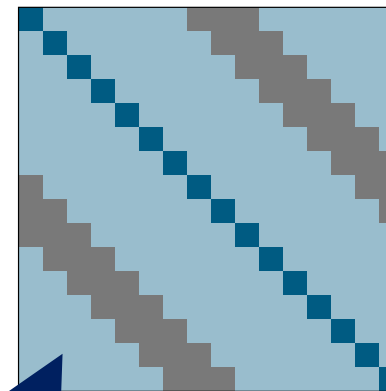
with truncation

Truncation of Green function elements

- Truncation of interactions of very far Voronoi cells
- Test system `long_cell_with_truncation` consists of a linear periodic chain of 16 atoms.
- Truncation switched on: `cutoff_radius = 6.9`



no truncation



with truncation

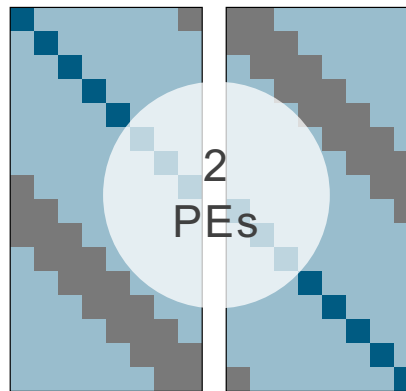
- Respect periodic boundary conditions

Truncation and Parallelization



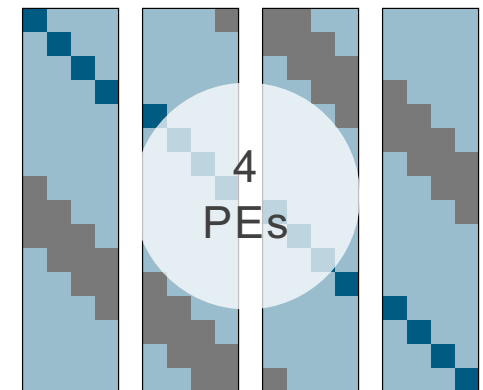
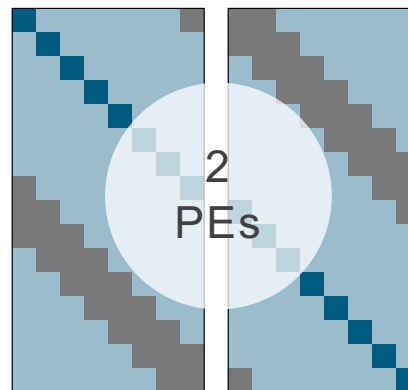
Truncation and Parallelization

- Parallelized using message passing interface (MPI)



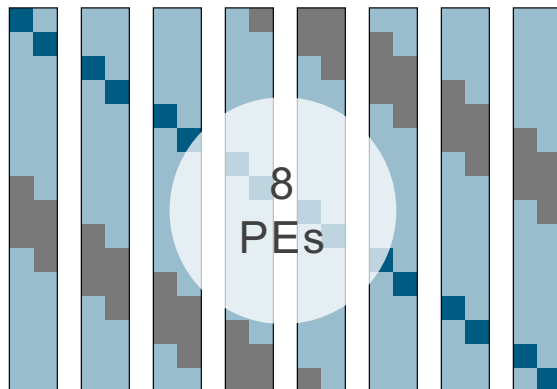
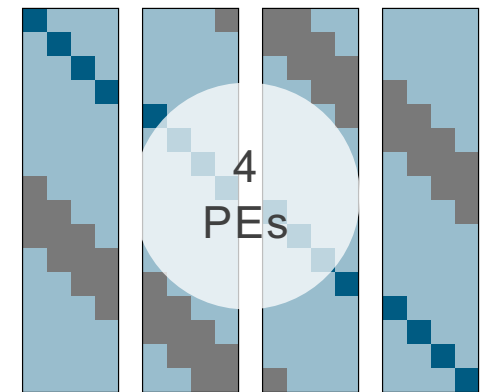
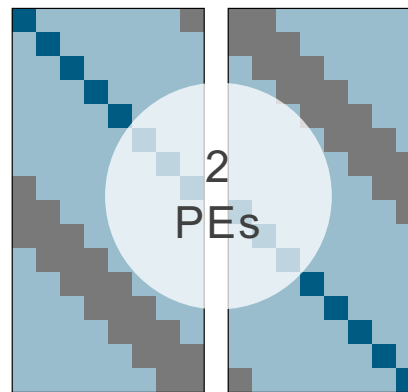
Truncation and Parallelization

- Parallelized using message passing interface (MPI)



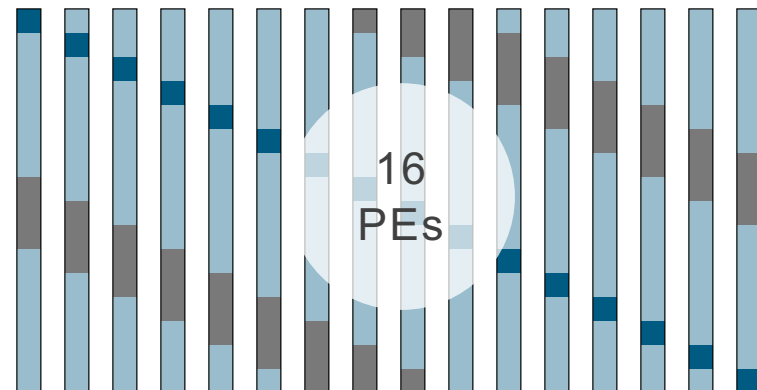
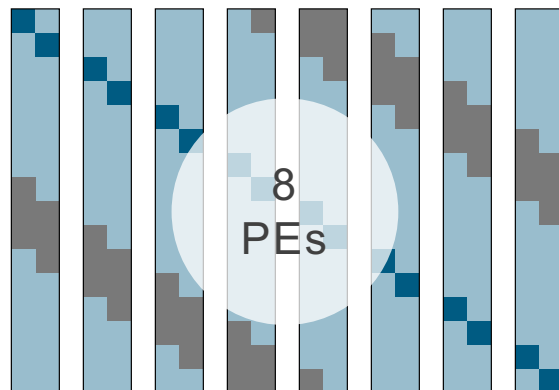
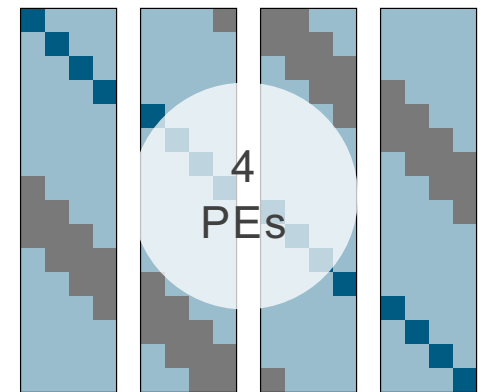
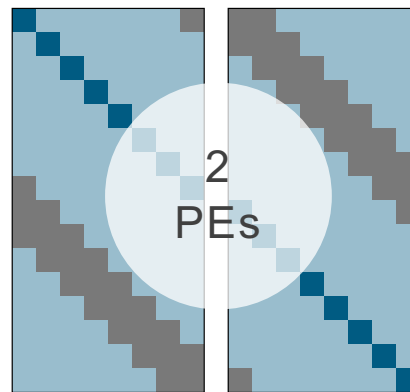
Truncation and Parallelization

- Parallelized using message passing interface (MPI)



Truncation and Parallelization

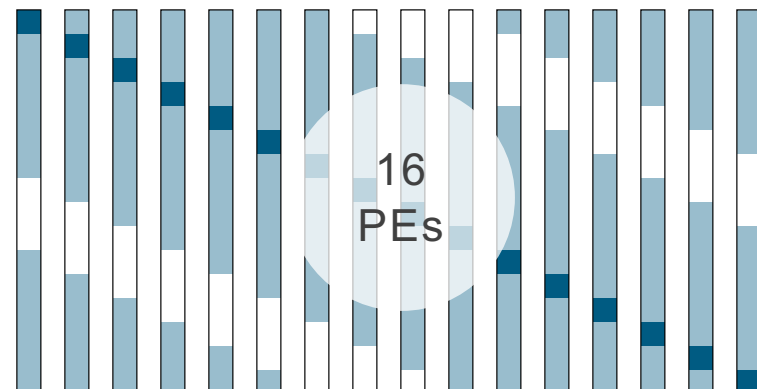
- Parallelized using message passing interface (MPI)



Truncation and Parallelization

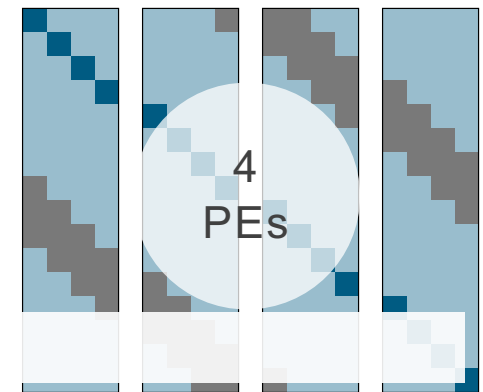
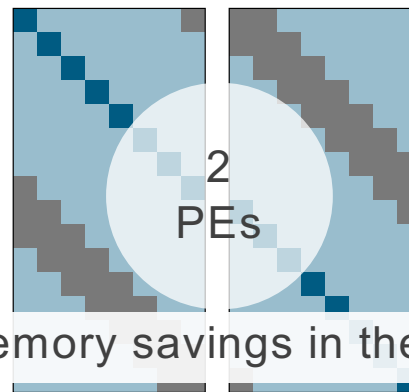


- Parallelized using message passing interface (MPI)
- Truncated Green function elements do not need to be stored (white) and processed in order to save memory, flops and bandwidth.

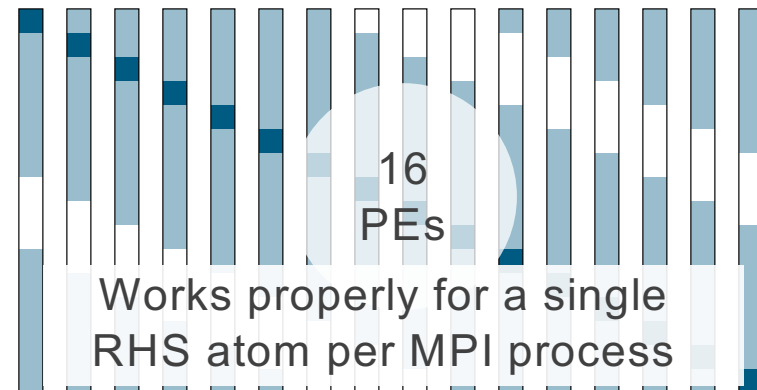
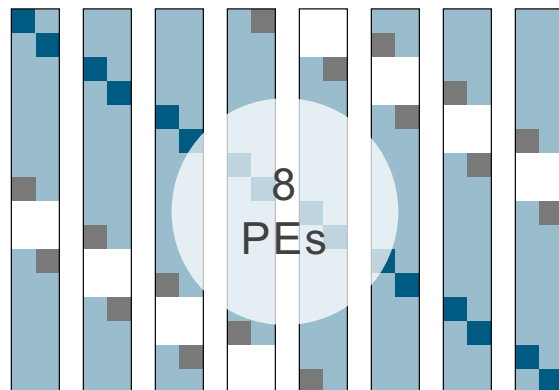


Truncation and Parallelization

- Parallelized using message passing interface (MPI)



No memory savings in these cases

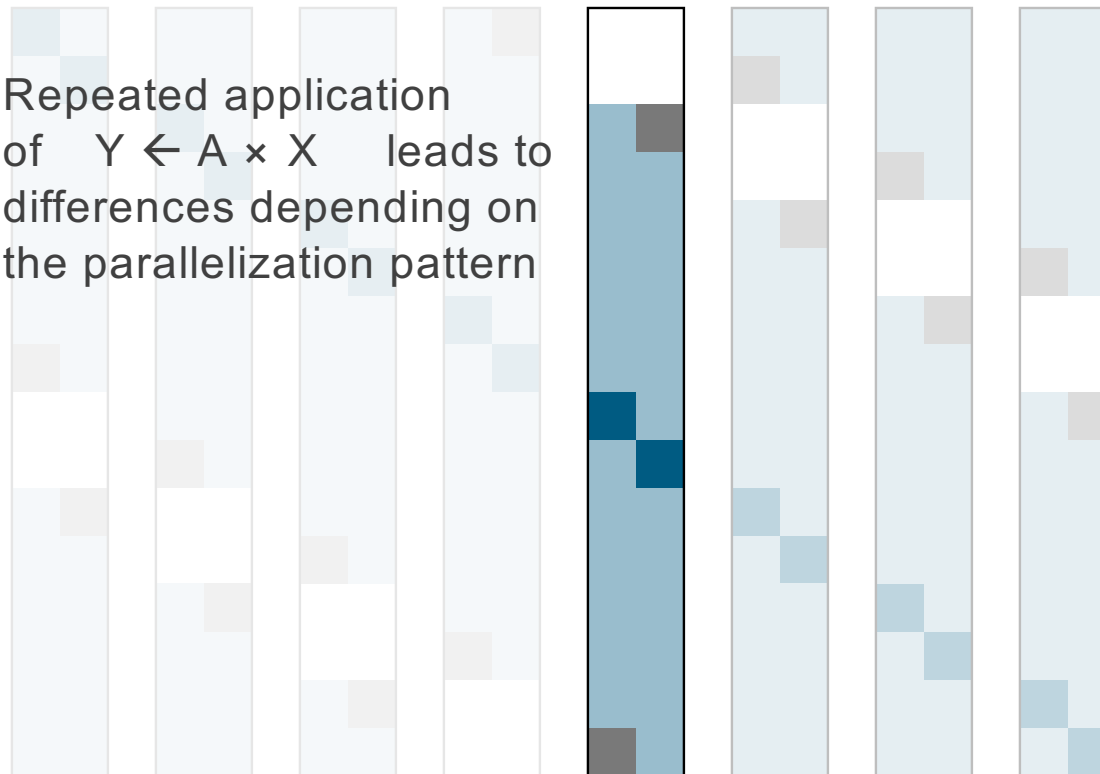


Works properly for a single
RHS atom per MPI process

Truncation and `num_local_atoms > 1`

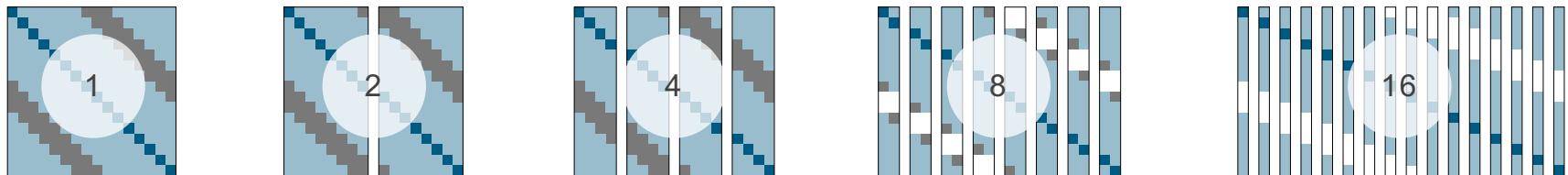
- 16 atoms treated on 8 MPI process elements

- Repeated application of $Y \leftarrow A \times X$ leads to differences depending on the parallelization pattern



Truncation and `num_local_atoms > 1`

- Effects becomes visible in charges and total energy



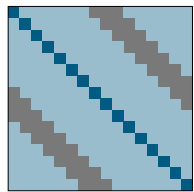
- Test system: `long_cell_with_truncation`

Parallelization	Total Energy (Ryd)	s-Charge #1
serial execution	-49575.87510476	.04182630
2 MPI processes	-49575.87510476	.04182630
4 MPI processes	-49575.87510476	.04182630
8 MPI processes	-49575.88989611	.04170299
16 MPI processes	-49575.91684984	.04190965

KKRnano – Truncation and Grouping

- Rectangular matrix X leads to storing truncated Green function elements

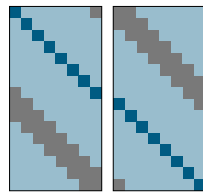
Current code status:



No

No

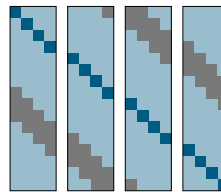
Yes



No

No

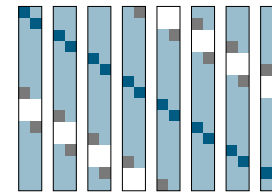
Yes



No

No

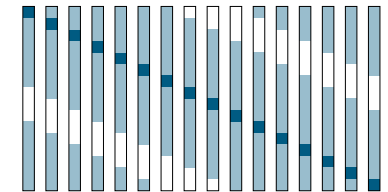
Ok



No

Ok

Ok



Yes

Yes

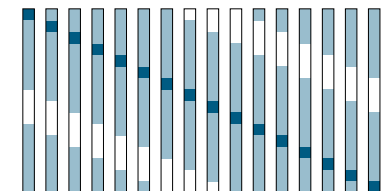
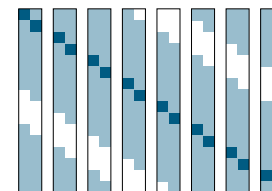
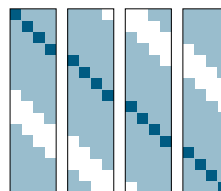
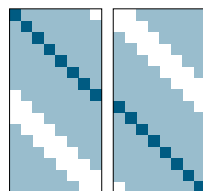
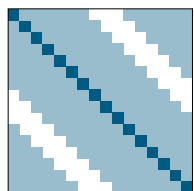
No

Correct

Memory

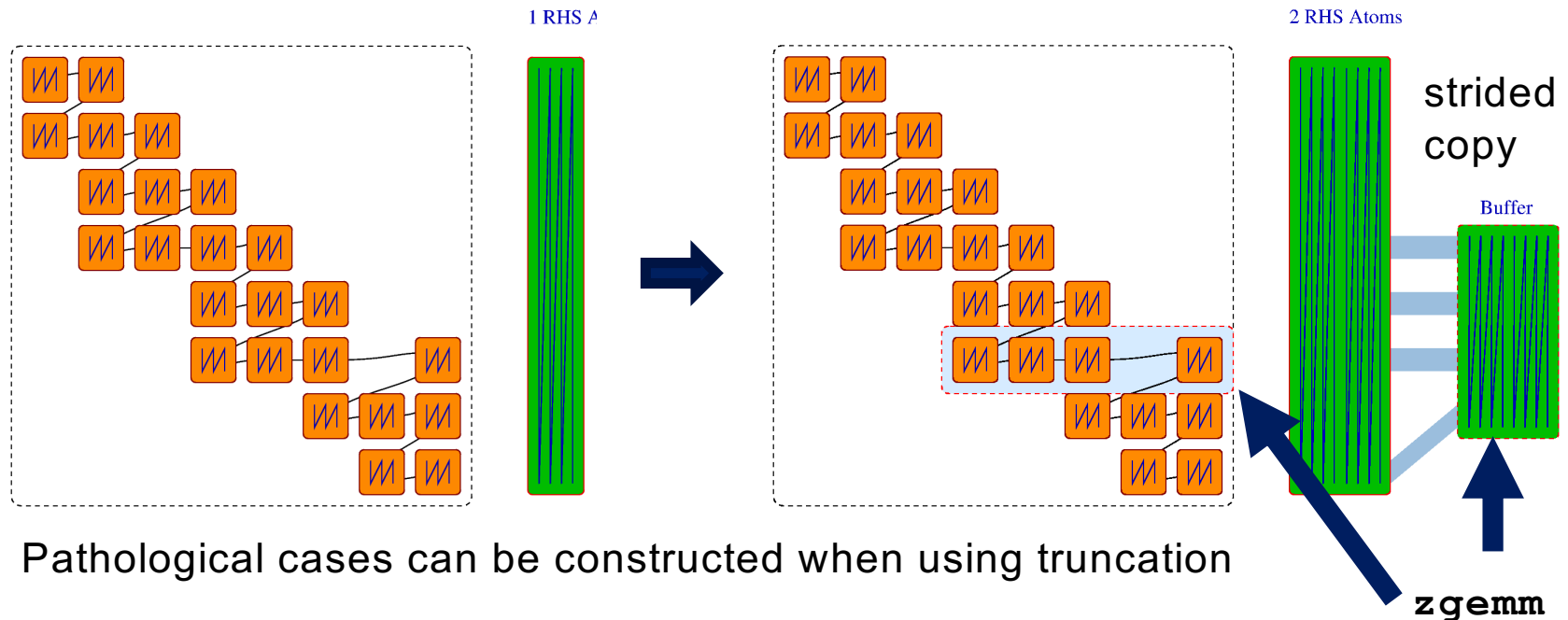
Efficient

Desired data layout:



KKRnano – Kernel Architecture

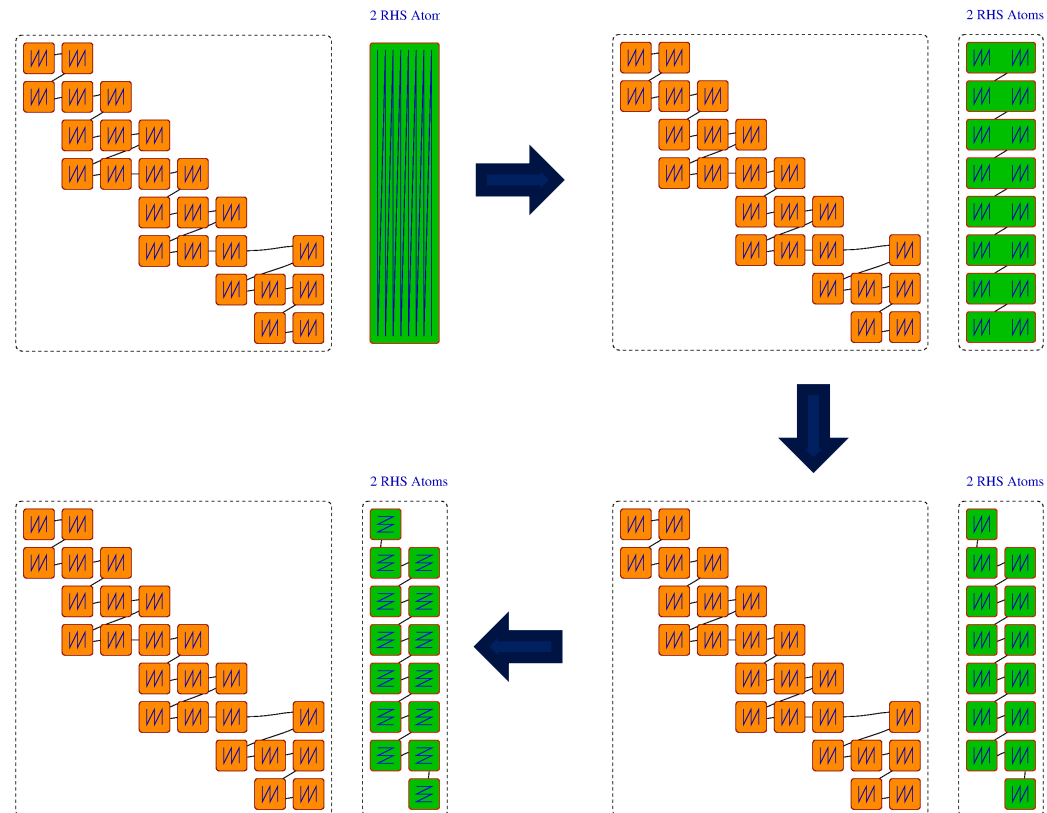
- Fast application of $A \times X$ required
- For the simplicity of bookkeeping, 1 atom per MPI process was fixed
- Needed to release constraint due to MPI overheads and arithmetic intensity



- Pathological cases can be constructed when using truncation

KKRnano - Kernel Restructuring

- Truncation leads to a different view of the operator A for each RHS atom
- Split long vectors of X into blocks
- Sparse storage of blocks of X in truncation mode
- Improved vectorization possible with transposed blocks of $X \rightarrow$ RowMajor

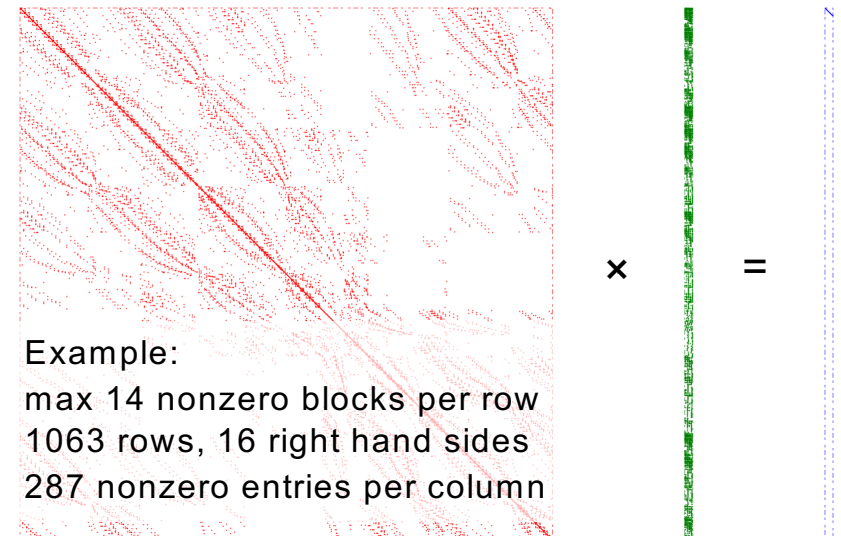
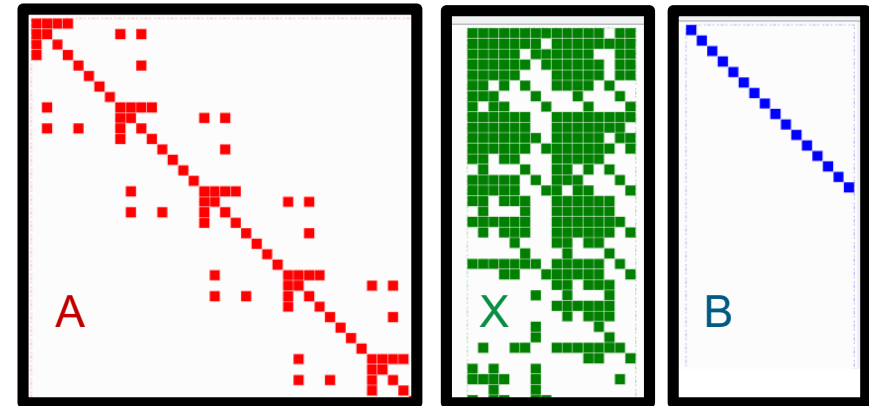


KKRnano – New Kernel Architecture

- Restructured KKRnano to use **BlockSparseRow** representations for the matrices A, B and X in

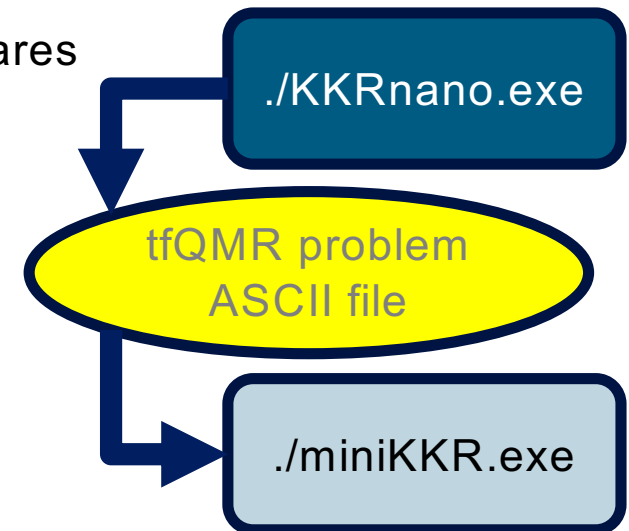
$$A \times X == B$$

- BSR-BSR multiplication $Y = A \times X$
planning performed ahead,
 $\text{shape}(Y) == \text{shape}(X)$
- Relies on the performance of **zgemm**
for many small block matrices,
usually 16×16 or 32×32 double
precision complex, hand written
version may outperform libraries



miniKKR – Kernel Performance

- miniKKR is the kernel of KKRnano extracted for performance engineering (~2000 Lines of Code)
- KKRnano exports a problem and its solution into a file
- miniKKR reads the problem, solves and compares
- Available under GPL v2 and/or MIT license
- Real world problem: MnGe (B20), 32×32 Matsubara pole at 1570 Kelvin,
14 nonzero blocks per row
287 nonzero blocks per column
16 atoms per MPI process



109 GFlop/s on a quadcore Xeon i7-6700 (3.7-3.8 GHz) → Eff. 45%
 324 GFlop/s on Xeon Phi KnL (64 core, SNC4, cache mode) → 14%

Arithmetic Intensity

Measures the ratio of computations over transferred data volume.

For matrix-matrix-multiplication

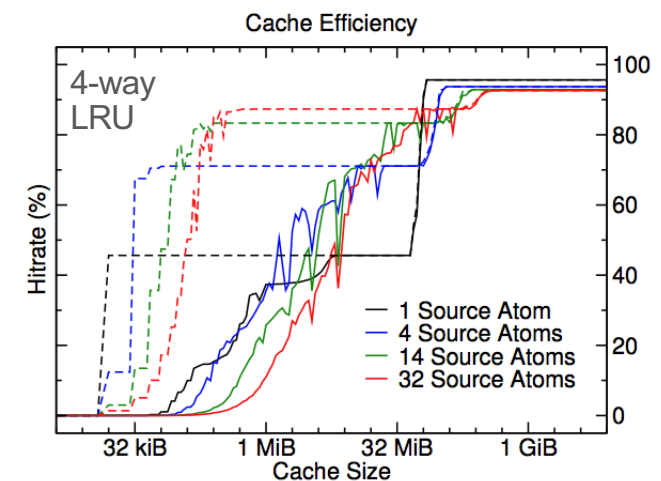
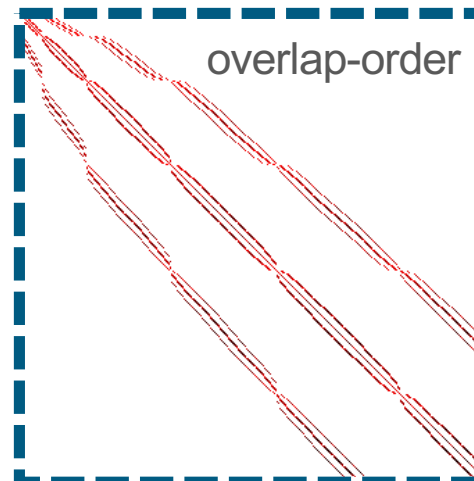
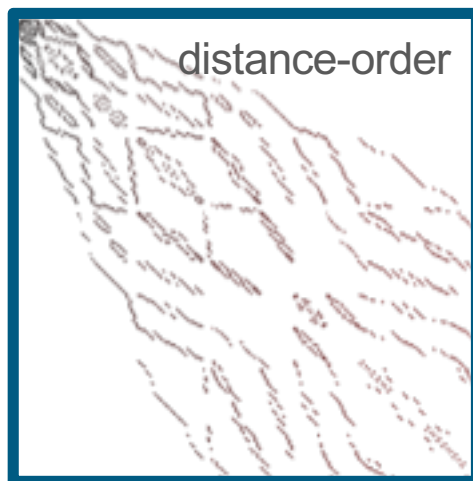
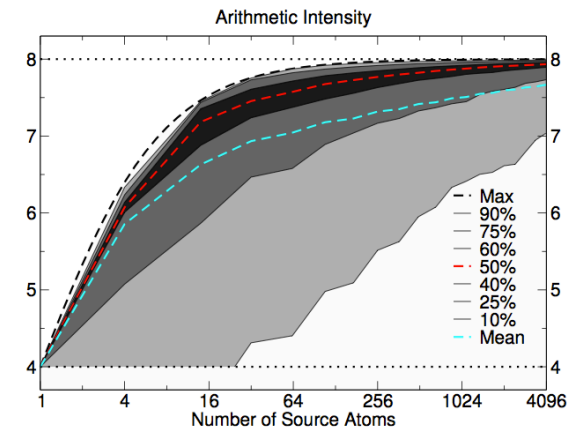
- $C = C + A \times B$
- Assume square matrices of dimension n
- $Al_{dgemm} = 2n^3 \text{ Flop} / (4n^2 * 8 \text{ Byte}) = n/16 \text{ Flop/Byte}$

FMA


(3x read, 1x write)
- $Al_{zgemm} = 8n^3 \text{ Flop} / (4n^2 * 16 \text{ Byte}) = n/8 \text{ Flop/Byte}, \quad Al_{32 \times 32} = 4.0 \text{ Flop/Byte}$
- Assumption that the 3 matrix operands fit into caches or registers

Data Locality Transformations

- Kernel performs sparse-matrix-vector with blocks
- Group the vectors of several source atoms to a matrix
- Sparse-matrix-matrix: Increase arithmetic intensity
- Default cache-blocking works only for dense matrices
- Reorder matrix entries to improve the cache efficiency



Summary

- Wave function based DFT needs $O(N^2)$ memory and $O(N^3)$ compute
- Green function based DFT is $O(N^2)$ or even $O(N)$ with truncation
- Restructuring of data layout for multiple scattering part of `KKRnano`
- Extraction of the mini-app `miniKKR` for performance engineering
- Tuning for current/future HPC platforms with many-core technology
- Searching for heuristics to maximize data reuse and save bandwidth

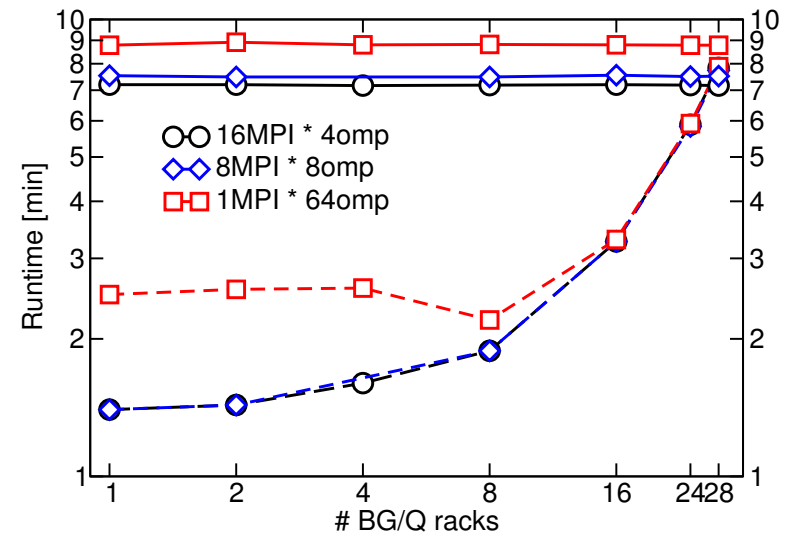
Outlook or ToDo-List for KKRnano

For the Green function solver:

- Implement GPU solver [1]
- Improve KnL performance ($< 20\%$)
- Preconditioning

Multipole summation technique for electrostatics scale $O(N^2_{\text{atoms}})$

- Fast multipole method?

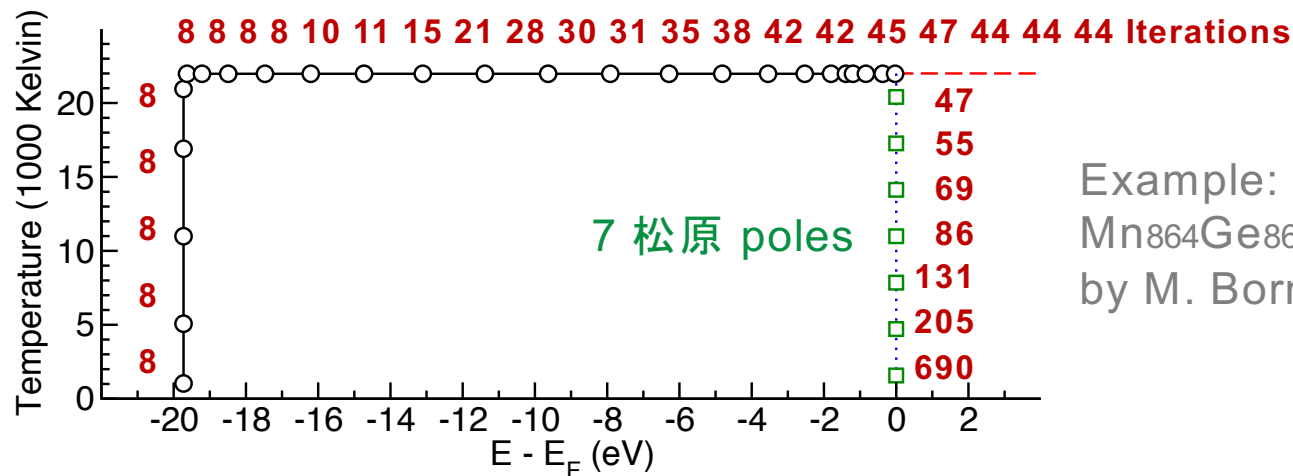


I/O formats and routines

[1] Addressing Materials Science Challenges Using GPU-accelerated POWER8 Nodes,
P. F. Baumeister, M. Bornemann, M. Bühler, Th. Hater, B. Krill, D. Pleiter, R. Zeller, EuroPar 2016 proceedings

Outlook - Preconditioning

KKRnano: State of the code: preconditioning is implemented, but no general efficient preconditioner available



Example:
Mn₈₆₄Ge₈₆₄ (B20)
by M. Bornemann

Idea: Use different truncation radii, find short ranged preconditioner by solving with $R_{\text{trunc}} = R_{\text{ref}}$, then MPI distribute data