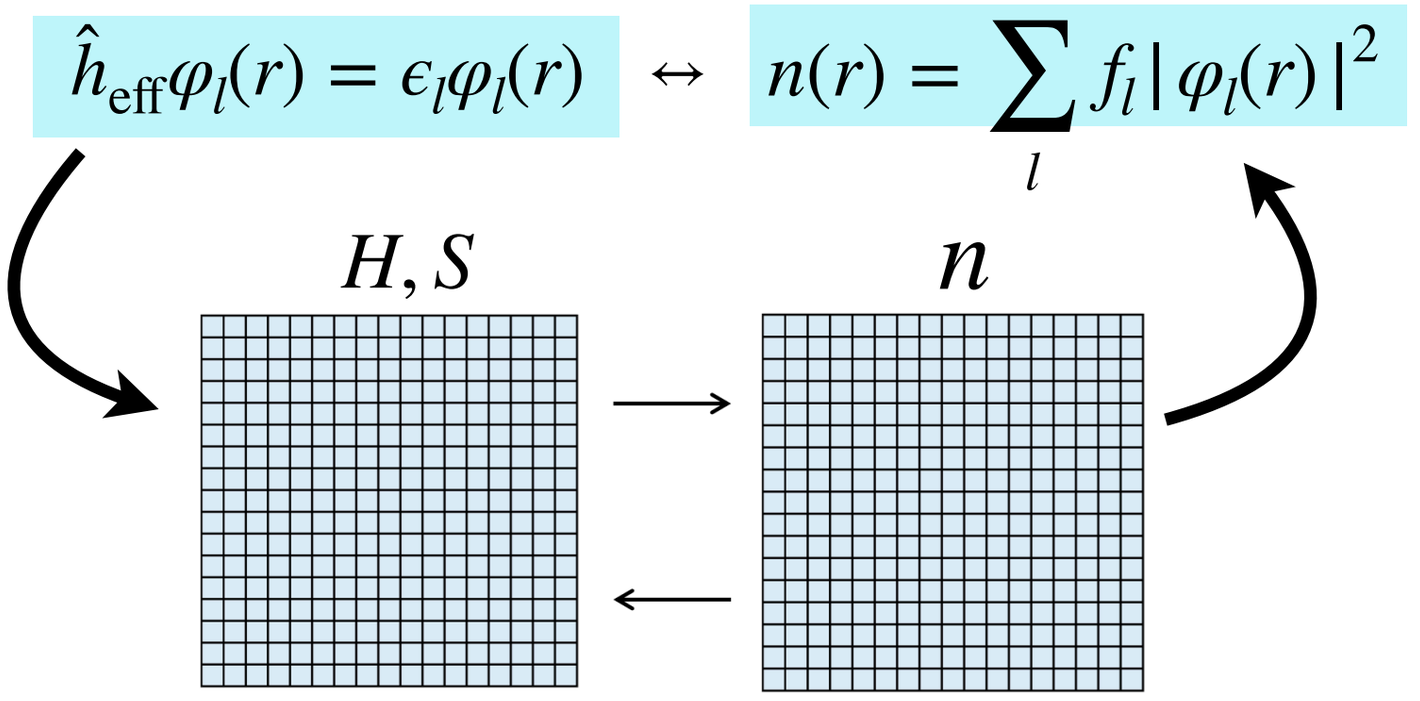
The ELSI Infrastructure
for Scalable Electronic Structure Theory

Victor Yu

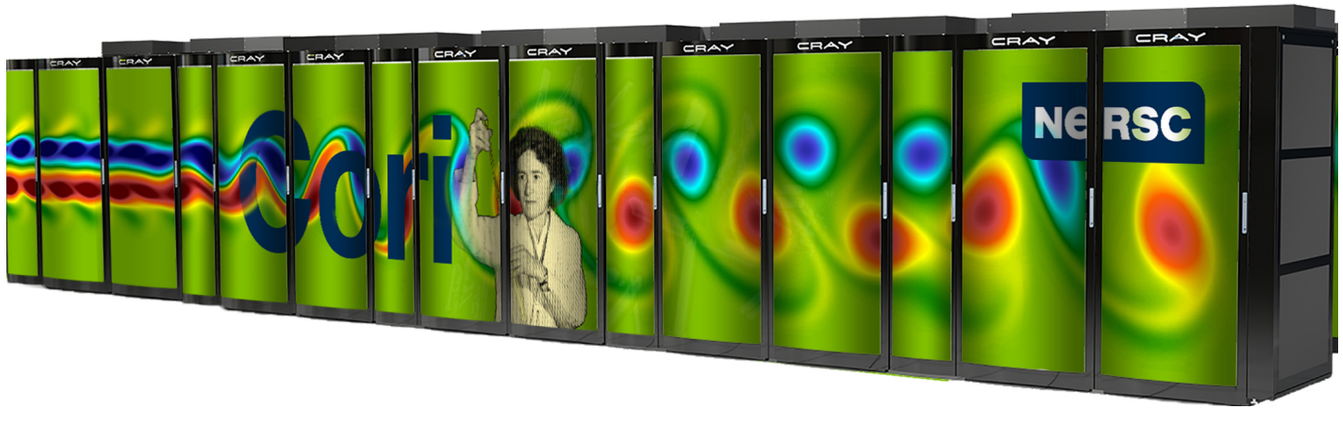
Victor Yu

Volker Blum

Duke University
Department of Mechanical Engineering and Materials Science
Durham, NC, USA
http://aims.pratt.duke.edu



https://www.elsi-interchange.org



http://www.nersc.gov/cori

ELSI - Acknowledgments

Nucleus: Emilio Artacho, 2014: *"Dear all, There will be a workshop in CECAM at Lausanne … aiming to kick-start an electronic structure library. … I hope you are interested"*

ELSI - Acknowledgments

Nucleus: Emilio Artacho, 2014: *"Dear all, There will be a workshop in CECAM at Lausanne … aiming to kick-start an electronic structure library. … I hope you are interested"*

NSF-SI2 - ACI-1450280:
Volker Blum, Jianfeng Lu, Lin Lin, Chao Yang, Alvaro Vazquez-Mayagoitia, Fabiano Corsetti

ELSI - Acknowledgments

Nucleus: Emilio Artacho, 2014: *"Dear all, There will be a workshop in CECAM at Lausanne … aiming to kick-start an electronic structure library. … I hope you are interested"*

NSF-SI2 - ACI-1450280:
Volker Blum, Jianfeng Lu, Lin Lin, Chao Yang, Alvaro Vazquez-Mayagoitia, Fabiano Corsetti

Why ELSI Works:

Victor Yu
(Duke)

Yingzhou Li
(Duke)

Will Huhn
(Duke)

William Dawson, Alberto Garcia, Ville Havu,
Ben Hourahine, Mathias Jacquelin, Weile Jia,
Murat Keceli, Raul Laasner, Björn Lange,
Wenhui Mi, Jonathan Moussa, Jose E. Roman,
Ali Seifitokaldani, Haizhao Yang;
ELPA, PEXSI, NTPoly, Slepc, …

ELSI - Acknowledgments

Nucleus: Emilio Artacho, 2014: *"Dear all, There will be a workshop in CECAM at Lausanne … aiming to kick-start an electronic structure library. … I hope you are interested"*

NSF-SI2 - ACI-1450280:
Volker Blum, Jianfeng Lu, Lin Lin, Chao Yang, Alvaro Vazquez-Mayagoitia, Fabiano Corsetti

Why ELSI Works:

Victor Yu (Duke)

Yingzhou Li (Duke)

Will Huhn (Duke)

William Dawson, Alberto Garcia, Ville Havu, Ben Hourahine, Mathias Jacquelin, Weile Jia, Murat Keceli, Raul Laasner, Björn Lange, Wenhui Mi, Jonathan Moussa, Jose E. Roman, Ali Seifitokaldani, Haizhao Yang; ELPA, PEXSI, NTPoly, Slepc, …

Electronic Structure Library:

Micael Oliveira, Yann Pouillon, Fabiano Corsetti, Nick Papior, many more.
https://esl.cecam.org
https://gitlab.com/ElectronicStructureLibrary

ELSI, Fdict, Futile, libfdf, libgridxc, libpsml, libxc, Psolver, pspio, xmlf90, …

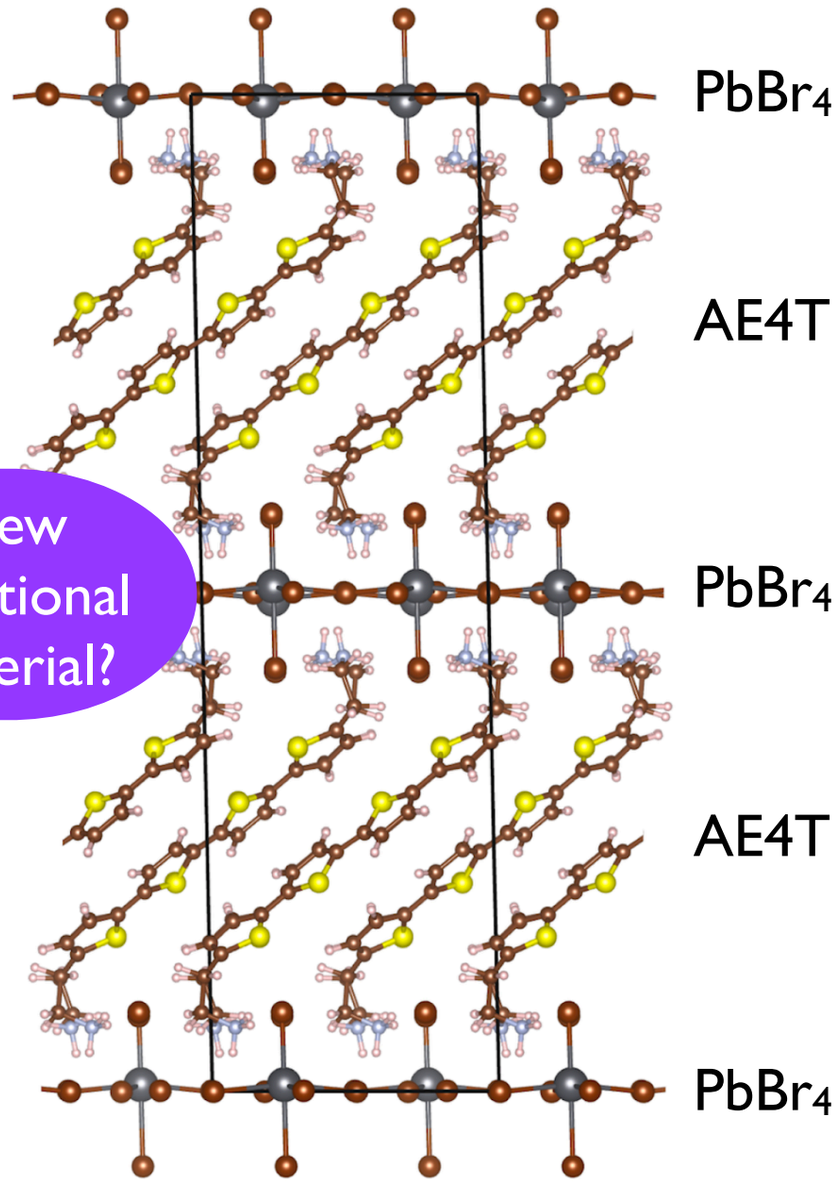
Materials Properties from Theory?

E.g., Organic-Inorganic Hybrid Semiconductors - Tailored Properties?

Component 1

New Functional Material?

Component 2



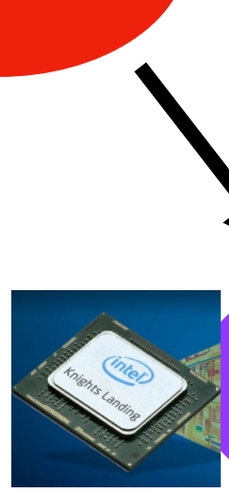
3D crystal
400+ atoms per unit cell

Liu, Huhn, Du, Vazquez-Mayagoitia, Dirkes, You, Kanai, Mitzi, Blum, Phys. Rev. Lett. ***121****, 146401 (2018)*

Materials Properties from Theory?

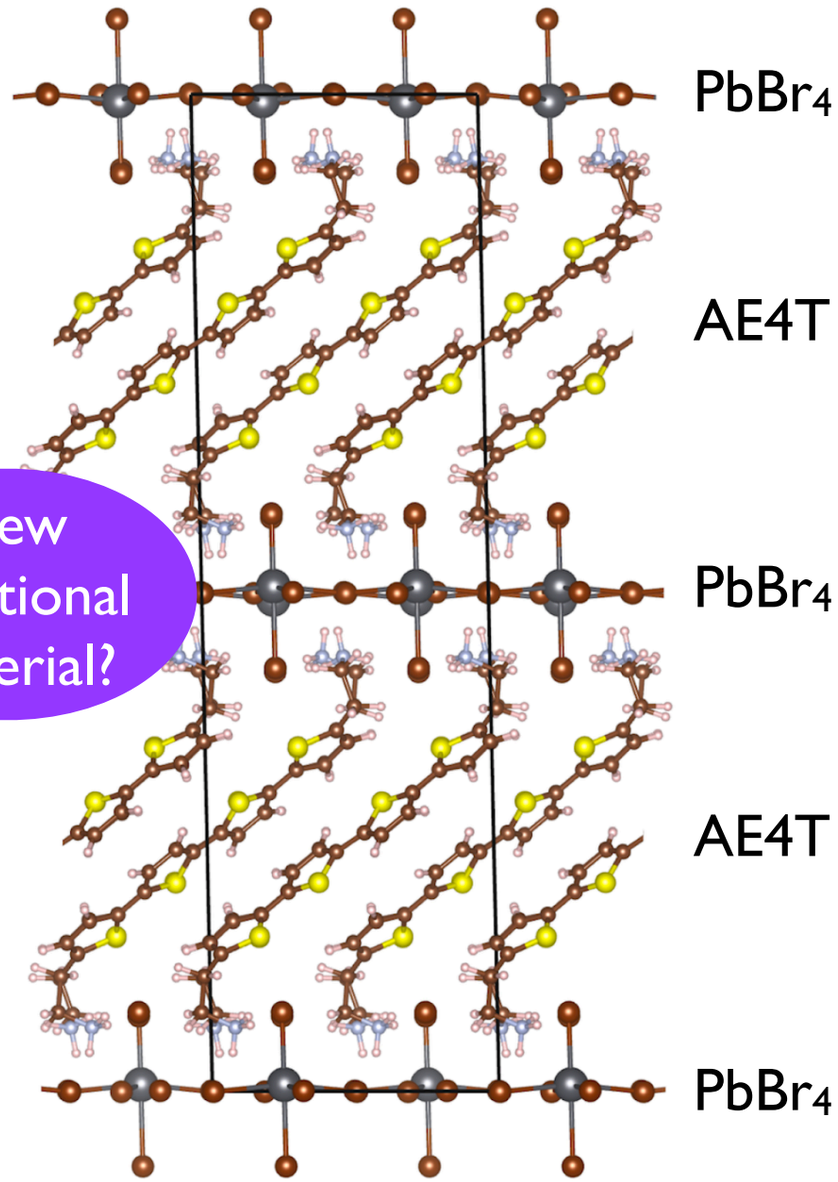
E.g., Organic-Inorganic Hybrid Semiconductors - Tailored Properties?

Component 1



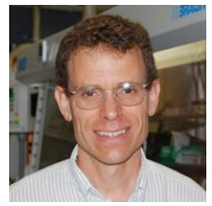
New Functional Material?

Component 2



3D crystal
400+ atoms per unit cell

Pathway 1: Make material, find out its properties.
Mitzi, Chondroudis, Kagan, Inorg. Chem. 38, 6246 (1999)



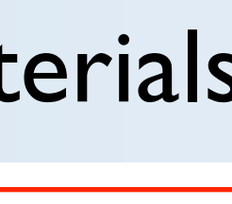
David Mitzi

Liu, Huhn, Du, Vazquez-Mayagoitia, Dirkes, You, Kanai, Mitzi, Blum, Phys. Rev. Lett. ***121****, 146401 (2018)*

Materials Properties from Theory?

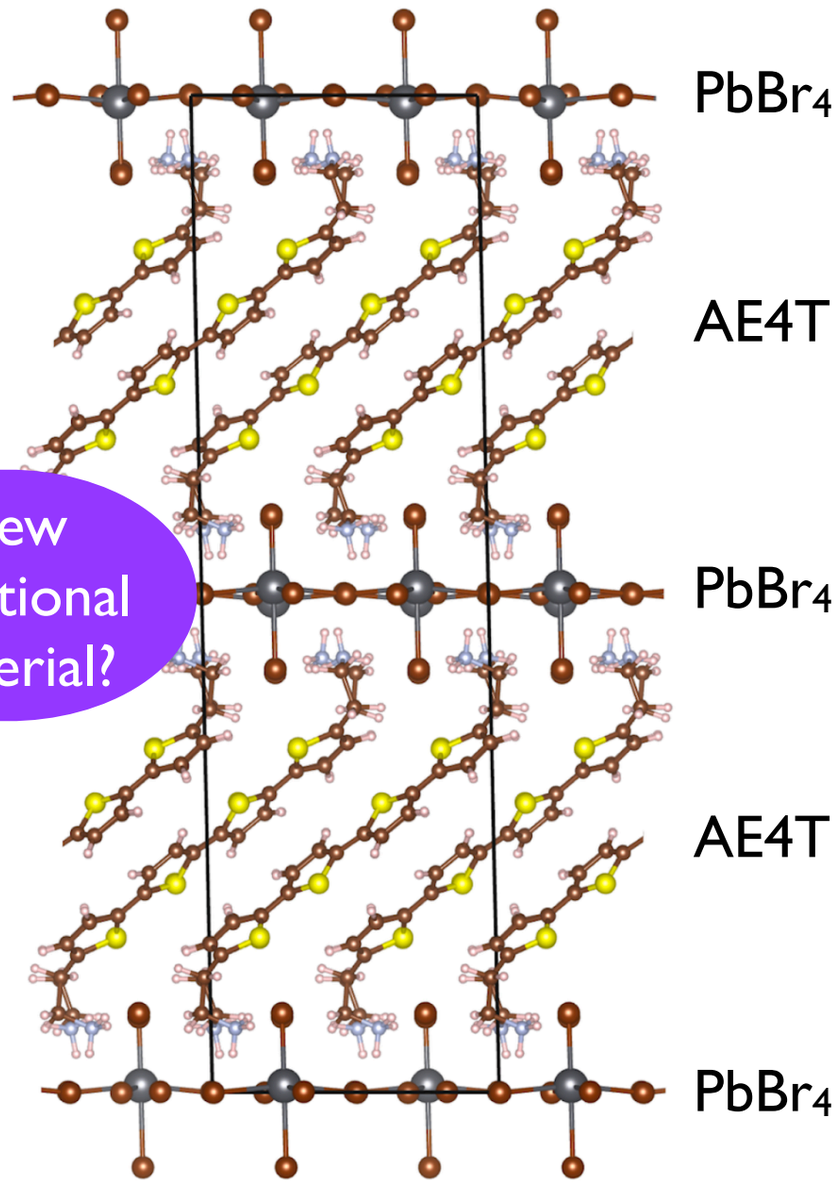
E.g., Organic-Inorganic Hybrid Semiconductors - Tailored Properties?

Component 1



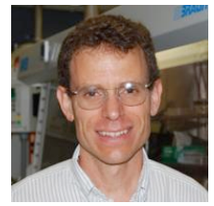
New Functional Material?

Component 2



3D crystal
400+ atoms per unit cell

Pathway 1: Make material, find out its properties.
Mitzi, Chondroudis, Kagan, Inorg. Chem. 38, 6246 (1999)



David Mitzi

Pathway 2: Accurate computational prediction?

$$\hat{\mathcal{H}}\Psi = E\Psi$$

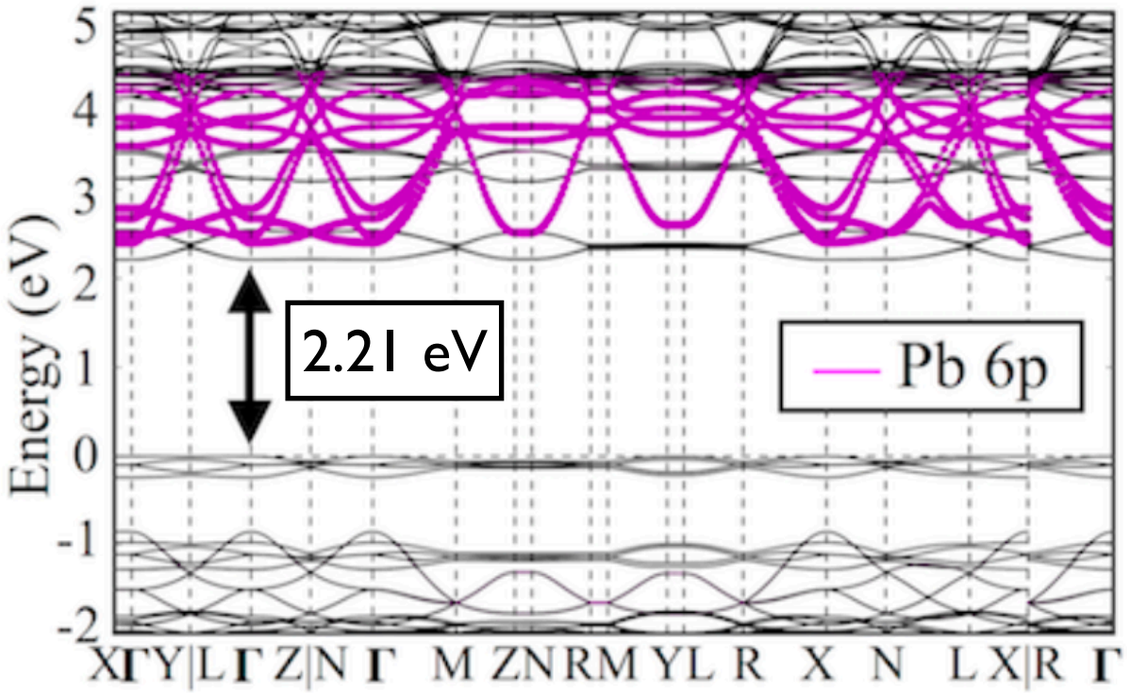
The theory we have is accurate enough.

But need to implement sufficiently good approximations on a real computer.

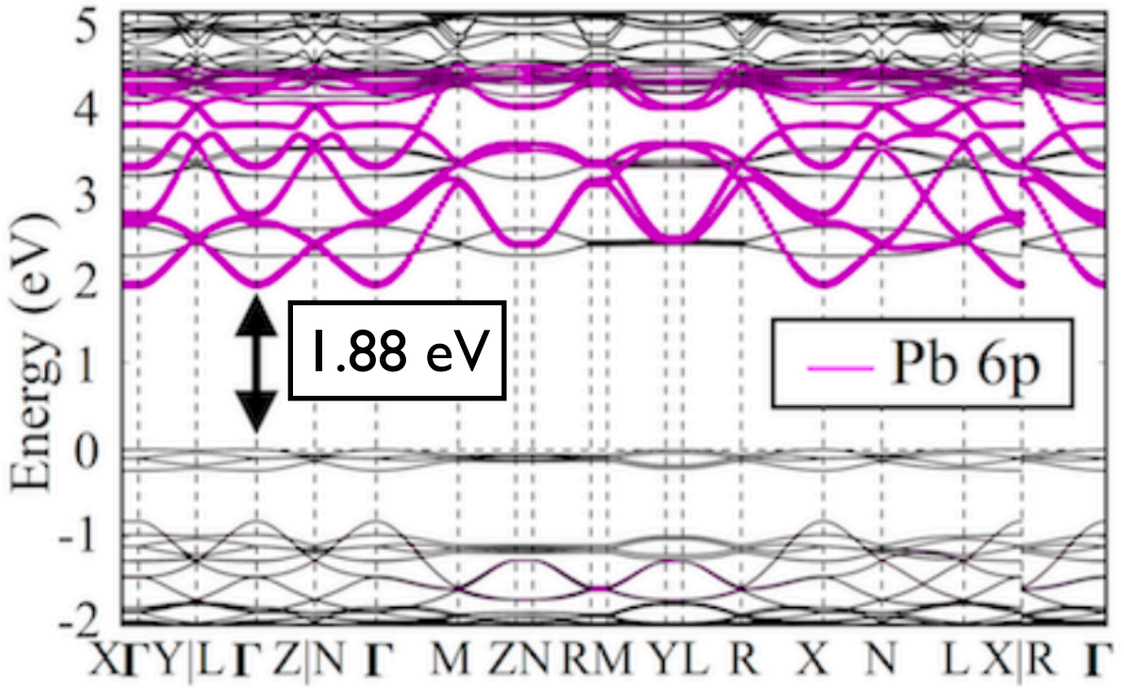
Liu, Huhn, Du, Vazquez-Mayagoitia, Dirkes, You, Kanai, Mitzi, Blum, Phys. Rev. Lett. ***121****, 146401 (2018)*

AE4T-$PbBr_4$: Energy Levels - Impact of Spin-Orbit Coupling

AE4T-$PbBr_4$ - HSE06, no SOC



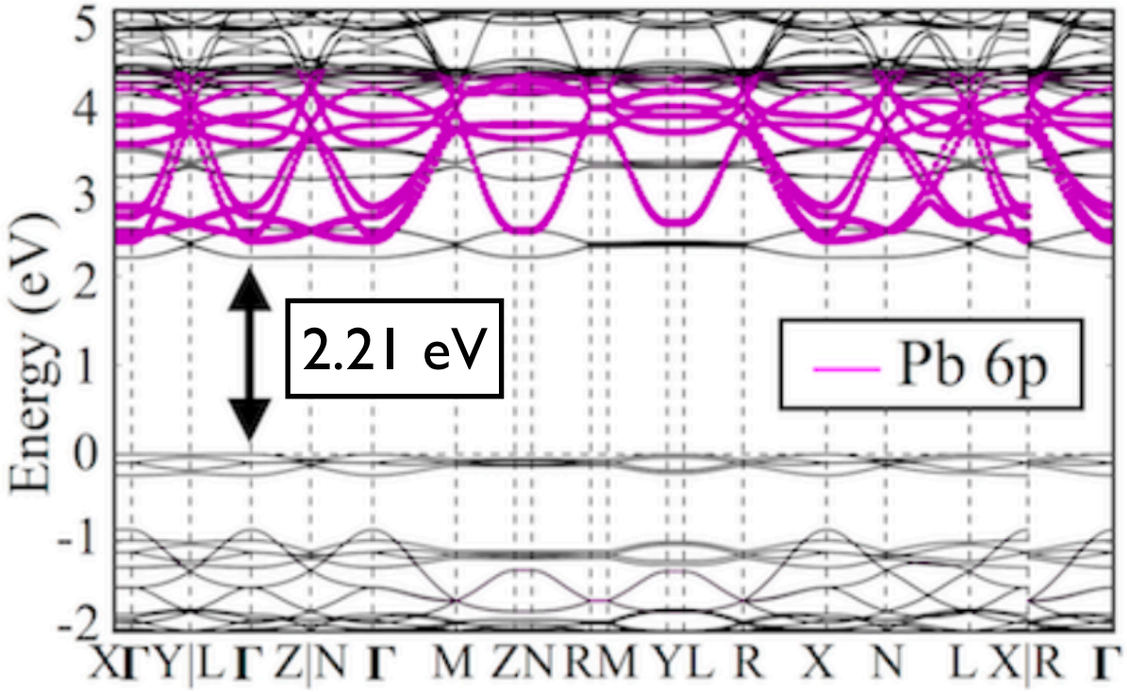
AE4T-$PbBr_4$ - HSE06, SOC



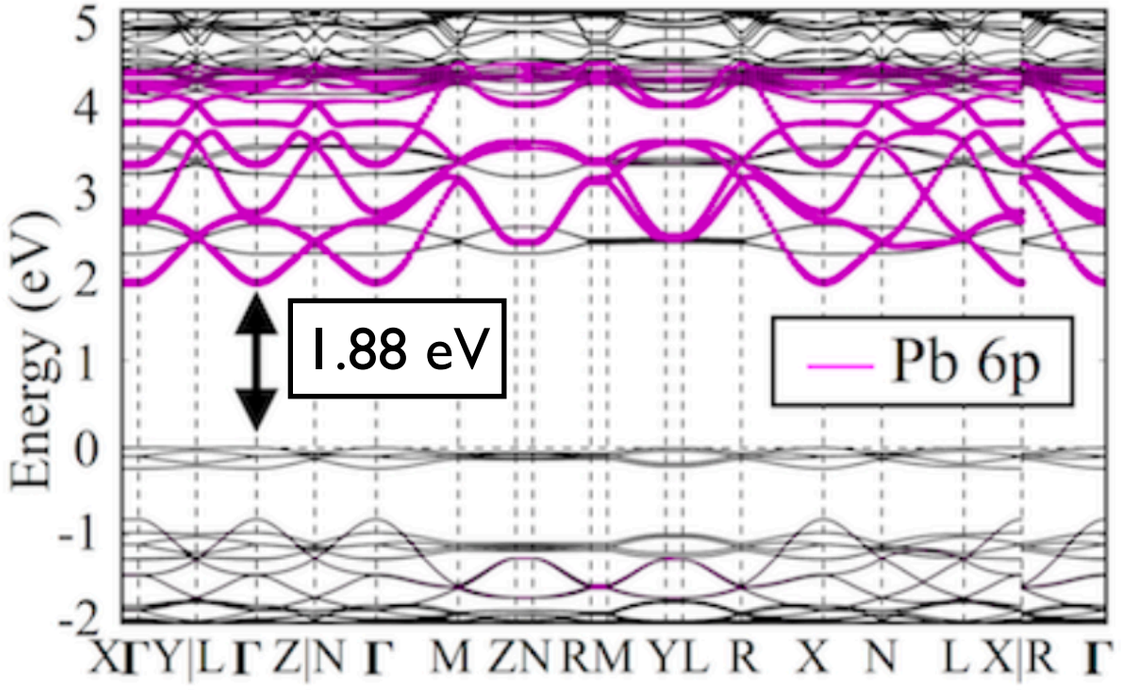
Liu, Huhn, Du, Vazquez-Mayagoitia, Dirkes, You, Kanai, Mitzi, Blum, Phys. Rev. Lett. ***121****, 146401 (2018)*

AE4T-$PbBr_4$: Energy Levels - Impact of Spin-Orbit Coupling

AE4T-$PbBr_4$ - HSE06, no SOC

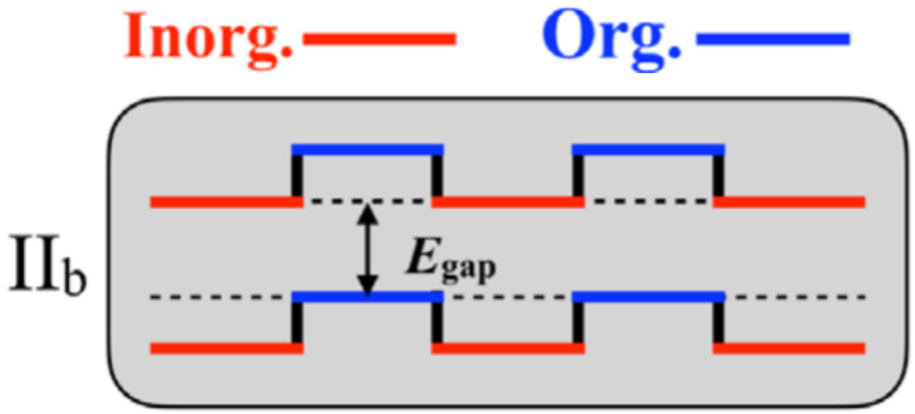


AE4T-$PbBr_4$ - HSE06, SOC



SOC changes the character of conduction band minimum ("electrons")

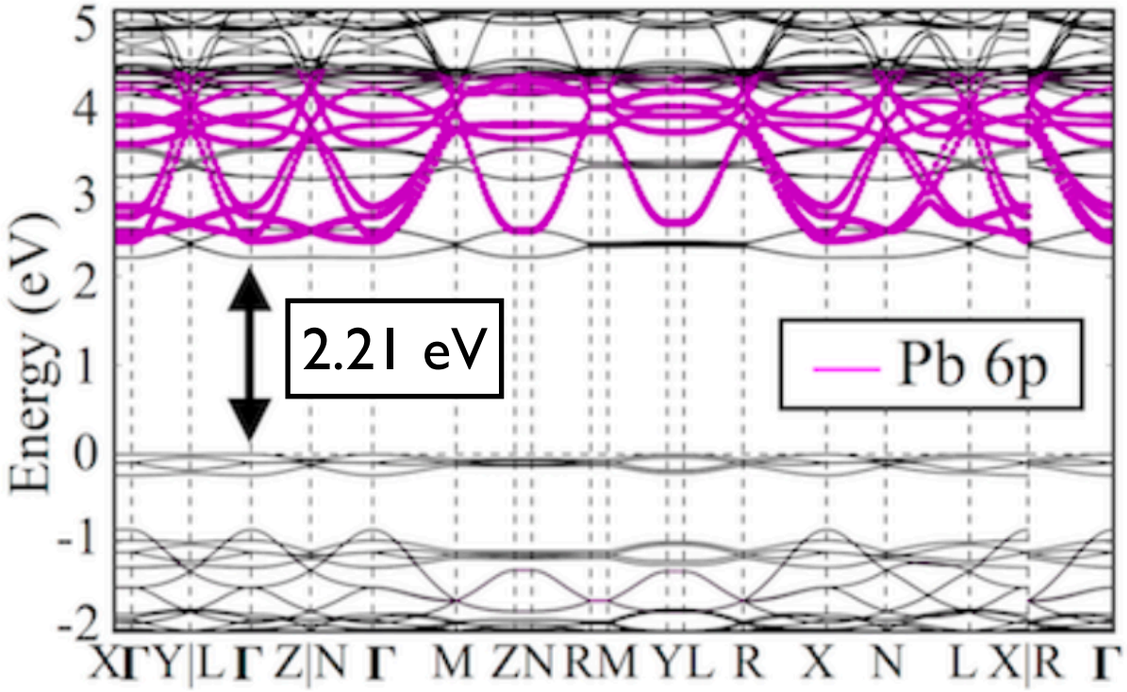
Holes on organic component, electrons on inorganic component: Type IIb Quantum Well



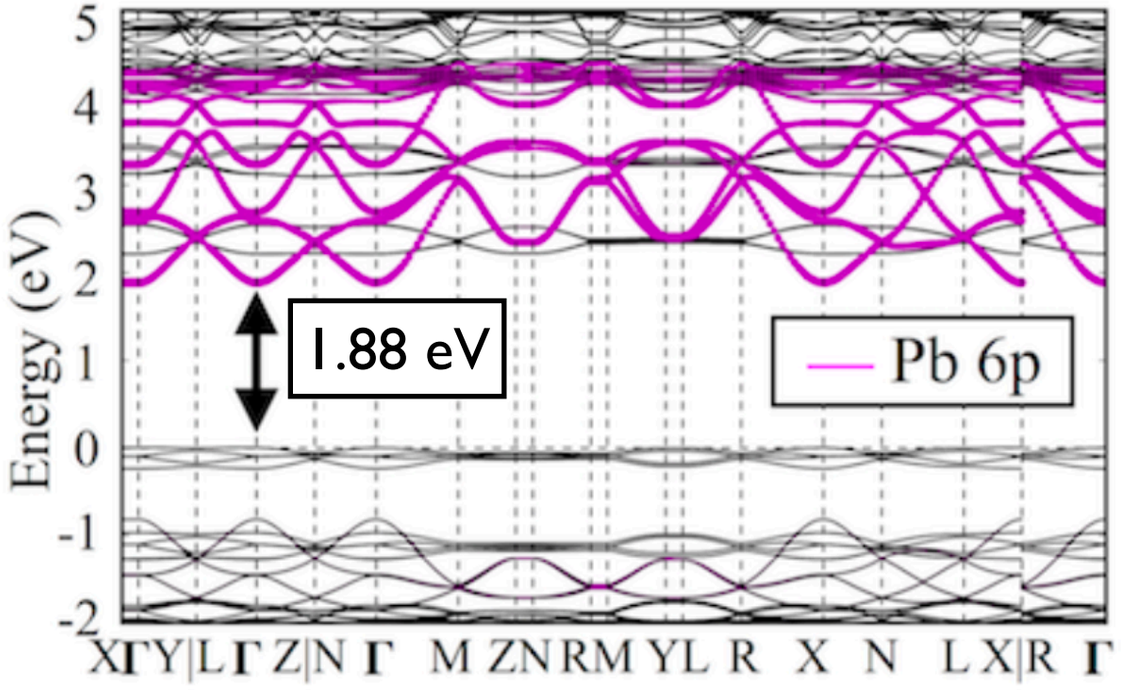
Liu, Huhn, Du, Vazquez-Mayagoitia, Dirkes, You, Kanai, Mitzi, Blum, Phys. Rev. Lett. ***121****, 146401 (2018)*

AE4T-$PbBr_4$: Energy Levels - Impact of Spin-Orbit Coupling

AE4T-$PbBr_4$ - HSE06, no SOC

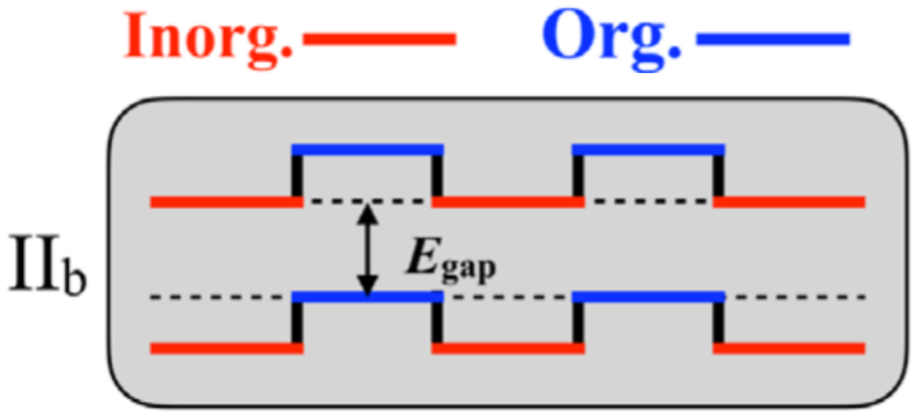


AE4T-$PbBr_4$ - HSE06, SOC



SOC changes the character of conduction band minimum ("electrons")

Holes on organic component, electrons on inorganic component: Type IIb Quantum Well

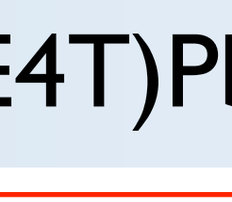


… but how exactly were these calculations run?

Liu, Huhn, Du, Vazquez-Mayagoitia, Dirkes, You, Kanai, Mitzi, Blum, Phys. Rev. Lett. ***121****, 146401 (2018)*

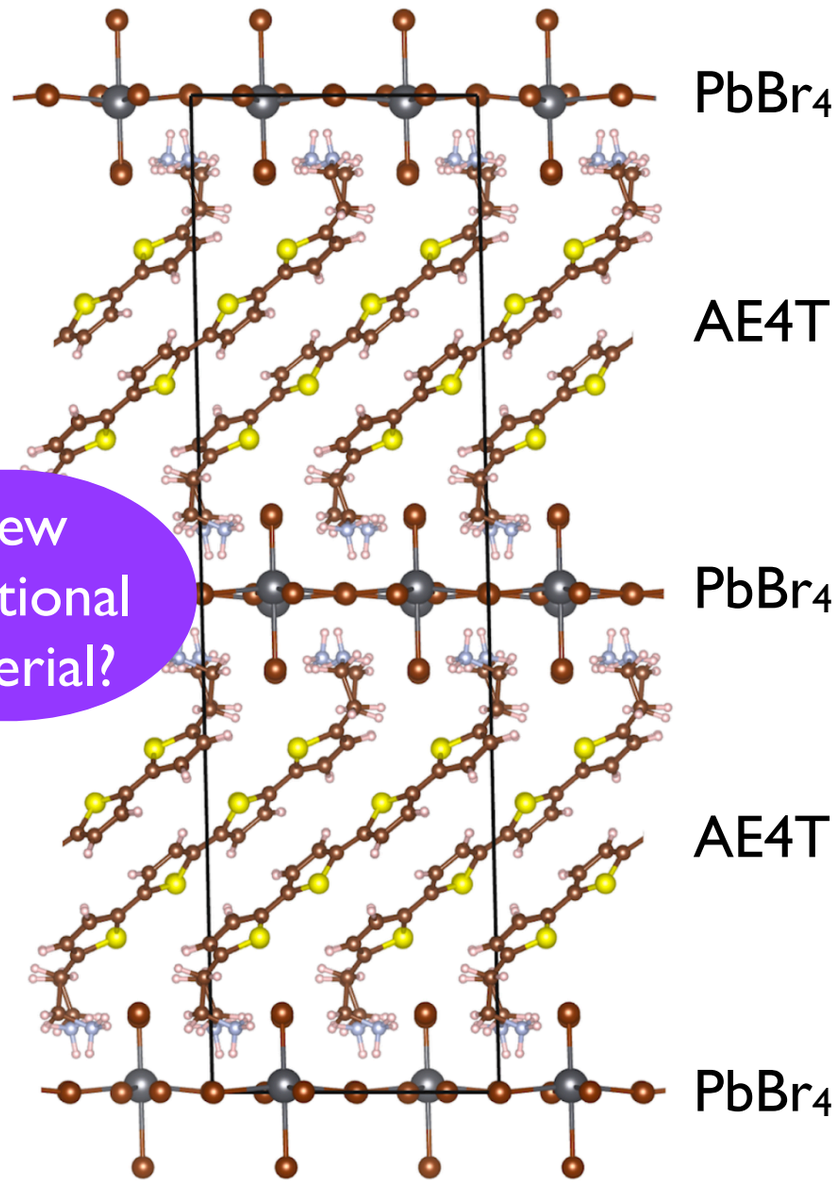
(AE4T)$PbBr_4$ - Computational Details

Component 1



New Functional Material?

Component 2



3D crystal
400+ atoms per unit cell

424 atoms (light and heavy)

Geometry:

DFT-PBE+vdW^{TS}
(Tkatchenko, Scheffler, PRL 2009)

tight settings
(12,384 basis functions)

several tens to hundreds of steps for geometry optimization

Electronic Structure:

DFT-HSE06+SOC

intermediate settings
(10,224 basis functions)

Liu, Huhn, Du, Vazquez-Mayagoitia, Dirkes, You, Kanai, Mitzi, Blum, Phys. Rev. Lett. ***121****, 146401 (2018)*

(AE4T)$PbBr_4$ - Computational Details

424 atoms (light and heavy)

Geometry:

DFT-PBE+vdW^{TS}

(Tkatchenko, Scheffler, PRL 2009)

tight settings
(12,384 basis functions)

several tens to hundreds of steps for geometry optimization

Electronic Structure:

DFT-HSE06+SOC

intermediate settings
(10,224 basis functions)

Liu, Huhn, Du, Vazquez-Mayagoitia, Dirkes, You, Kanai, Mitzi, Blum, Phys. Rev. Lett. ***121****, 146401 (2018)*

(AE4T)$PbBr_4$ - Computational Details

Which computer to use?

424 atoms (light and heavy)

Geometry:

DFT-PBE+vdW^{TS}

(Tkatchenko, Scheffler, PRL 2009)

tight settings
(12,384 basis functions)

several tens to hundreds of steps for geometry optimization

Electronic Structure:

DFT-HSE06+SOC

intermediate settings
(10,224 basis functions)

Liu, Huhn, Du, Vazquez-Mayagoitia, Dirkes, You, Kanai, Mitzi, Blum, Phys. Rev. Lett. ***121****, 146401 (2018)*

(AE4T)$PbBr_4$ - Computational Details

<u>Which computer to use?</u>

https://www.alcf.anl.gov/theta

Intel KNL
9.8 PFlops (231,424 cores)

1,024 - 2,048 cores
comfortably perform these tasks

<u>424 atoms (light and heavy)</u>

<u>Geometry:</u>

DFT-PBE+vdW^{TS}
(Tkatchenko, Scheffler, PRL 2009)

tight settings
(12,384 basis functions)

several tens to hundreds of steps for geometry optimization

<u>Electronic Structure:</u>

DFT-HSE06+SOC

intermediate settings
(10,224 basis functions)

Liu, Huhn, Du, Vazquez-Mayagoitia, Dirkes, You, Kanai, Mitzi, Blum, Phys. Rev. Lett. ***121****, 146401 (2018)*

However, How to Go Much Larger?

1. Real space grid operations

$$h_{ij} = \int d^3r \varphi_i(\boldsymbol{r}) \hat{h}_{\mathrm{KS}} \varphi_j(\boldsymbol{r})$$

Basis functions, Hamiltonian, Kohn-Sham potential etc.

2. Matrix algebra (basis space)

$$\underline{\underline{h}}\,\underline{c}_k = \epsilon_k \underline{\underline{s}}\,\underline{c}_k$$

Kohn-Sham eigenvalue problem

However, How to Go Much Larger?

1. Real space grid operations

$$h_{ij} = \int d^3r \varphi_i(\boldsymbol{r}) \hat{h}_{\mathrm{KS}} \varphi_j(\boldsymbol{r})$$

Basis functions, Hamiltonian, Kohn-Sham potential etc.

2. Matrix algebra (basis space)

$$\underline{\underline{h}}\,\underline{c}_k = \epsilon_k\,\underline{\underline{s}}\,\underline{c}_k$$

Kohn-Sham eigenvalue problem

- Large "prefactor:" Dominant for standard problems
- Mature algorithms (Delley, others)
- *O*(*N*) scalability possible in all steps
- *relatively* simple parallelization

V. Havu, V. Blum, P. Havu, M. Scheffler, J. Comp. Phys. ***228****, 8367-8379 (2009)*

However, How to Go Much Larger?

1. Real space grid operations

$$h_{ij} = \int d^3r \varphi_i(\boldsymbol{r}) \hat{h}_{\mathrm{KS}} \varphi_j(\boldsymbol{r})$$

Basis functions, Hamiltonian, Kohn-Sham potential etc.

- Large "prefactor:" Dominant for standard problems
- Mature algorithms (Delley, others)
- *O*(*N*) scalability possible in all steps
- *relatively* simple parallelization

V. Havu, V. Blum, P. Havu, M. Scheffler, J. Comp. Phys. ***228****, 8367-8379 (2009)*

2. Matrix algebra (basis space)

$$\underline{\underline{h}}\,\underline{c}_k = \epsilon_k \underline{\underline{s}}\,\underline{c}_k$$

Kohn-Sham eigenvalue problem

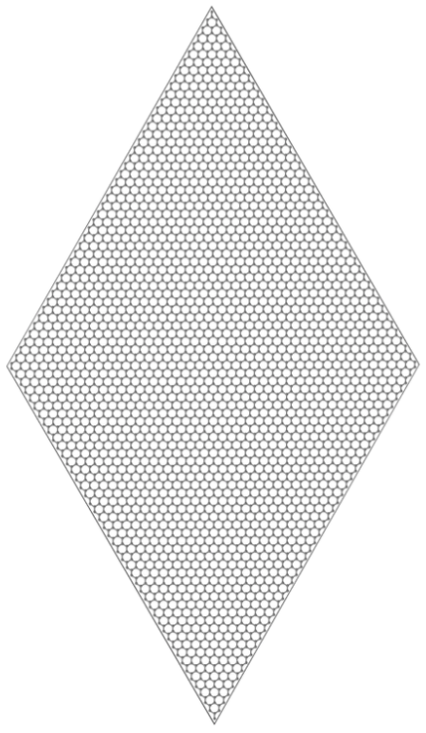
"Conventional" solvers (Lapack-like):

- Small prefactor for NAO's: affordable up to ≥1,000 atoms
- Robust, general (metals!)
- *O*(size^3) scalability inevitable
- Massively parallel scalability not out of the box

How far can we push such solvers?

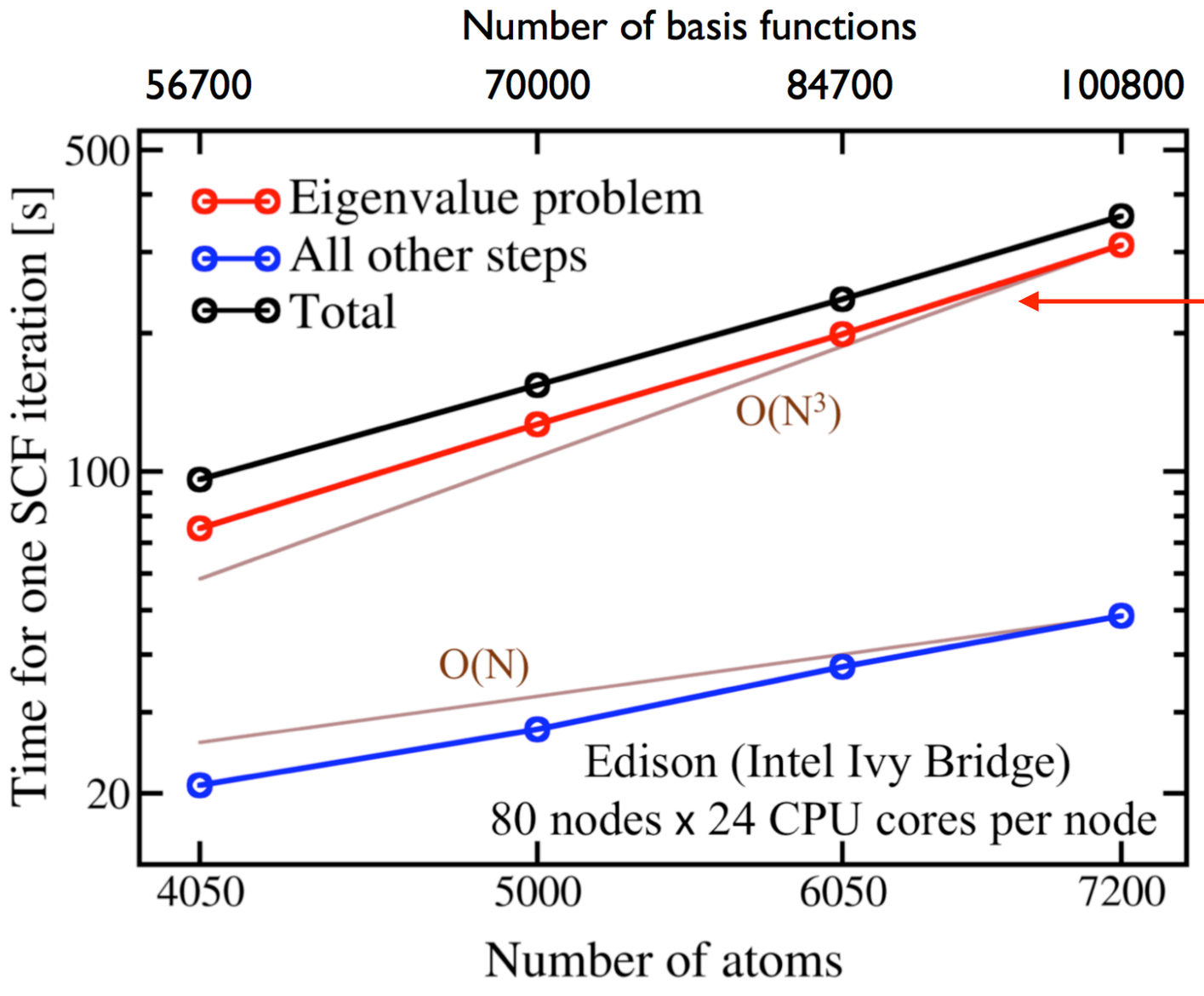
Typical Scaling - $O(N^3)$ Wall

$$H\varphi = \epsilon S\varphi$$



Graphene monolayer,
4050 atoms

FHI-aims, PBE, "light" settings



ELPA Library
http://elpa.rzg.mpg.de

Generic problem for any Kohn-Sham DFT code ... solution strategies?

In Principle, Much Larger Systems Are Possible

IOP PUBLISHING JOURNAL OF PHYSICS: CONDENSED MATTER

J. Phys.: Condens. Matter **22** (2010) 074207 (6pp) doi:10.1088/0953-8984/22/7/074207

Calculations for millions of atoms with density functional theory: linear scaling shows its potential

D R Bowler[1,2,3] and T Miyazaki[4]

In Principle, Much Larger Systems Are Possible

IOP PUBLISHING
JOURNAL OF PHYSICS: CONDENSED MATTER
J. Phys.: Condens. Matter **22** (2010) 074207 (6pp)
doi:10.1088/0953-8984/22/7/074207

Calculations for millions of atoms with density functional theory: linear scaling shows its pote

D R Bowler[1,2,3] and T Miyazaki

JCTC Journal of Chemical Theory and Computation
Artic
pubs.acs.org/J

Hybrid MPI-OpenMP Parallelism in the ONETEP Linear-Scaling Electronic Structure Code: Application to the Delamination of Cellulose Nanofibrils

Karl A. Wilkinson,[†] Nicholas D. M. Hine,[‡] and Chris-Kriton Skylaris*,[†]

JCTC Journal of Chemical Theory and

Linear Scaling Self-Consistent Field Calculations with Millions of Atoms in the Condensed Phase

Joost VandeVondele,*,[†] Urban Borštnik,[‡,§] and Jürg Hutter[‡]

[†]Department of Materials, ETH Zurich, Wolfgang-Pauli-Strasse 27, 8093 Zurich, Switzerland
[‡]Physical Chemistry Institute, University of Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland

ABSTRACT: In this work, the applicability and performance of a linear scaling algorithm is investigated for three-dimensional condensed phase systems. A simple but robust approach based on the matrix sign function is employed together with a thresholding matrix multiplication that does not require a prescribed sparsity pattern. Semiempirical methods and density functional theory have been tested. We demonstrate that self-consistent calculations with 1 million atoms are feasible for simple systems. With this approach, the computational cost of the calculation depends strongly on basis set quality. In the current

• • •

In Principle, Much Larger Systems Are Possible

IOP Publishing
J. Phys.: Condens. Matter **22** (2010) 074207 (6pp)
Journal of Physics: Condensed Matter
doi:10.1088/0953-8984/22/7/074207

Calculations for millions of atoms with density functional theory: linear scaling shows i

D R Bowler[1,2,3]

However:

- System- and/or density-functional specific algorithms
- The optimal algorithms changes from small to large scale
- Complex, algorithm-specific parallelization (matrix storage distribution, communication patterns)
- Very different supercomputing paradigms
  - must be addressed at low level

→ seamless, easy-to-use, general implementation from laptop to supercomputer not straightforward

pubs.acs.org/J

r-Scaling
nation of

JCT

Linear S
Atoms ir

Joost Vande
[†]Department of
[‡]Physical Chemi

ABSTRACT:
condensed phase systems. A simple but robust approach based on the matrix sign function is employed together with a thresholding matrix multiplication that does not require a prescribed sparsity pattern. Semiempirical methods and density functional theory have been tested. We demonstrate that self-consistent calculations with 1 million atoms are feasible for simple systems. With this approach, the computational cost of the calculation depends strongly on basis set quality. In the current

• • •

Agenda for the Remainder of this Talk

- Making an Eigenvalue Solver Scalable (ELPA library)
- Bypassing the Eigenvalue Problem - Density-Matrix Based Approaches
- ELSI Infrastructure

A Massively Parallel Dense Eigensolver: "ELPA"

$$\underline{\underline{h}}\,\underline{c}_k = \epsilon_k\,\underline{\underline{s}}\,\underline{c}_k$$

Given a matrix H and metric S (dimension *N*),
find *M* eigenvalue/eigenvector pairs ε_k/c_k

http://elpa.rzg.mpg.de

A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H.-J. Bungartz, H. Lederer,
The Journal of Physics: Condensed Matter 26, 213201 (2014).

A Massively Parallel Dense Eigensolver: "ELPA"

$$\underline{\underline{h}}\,\underline{c}_k = \epsilon_k\,\underline{\underline{s}}\,\underline{c}_k$$

Given a matrix H and metric S (dimension *N*), find *M* eigenvalue/eigenvector pairs ε_k/c_k

<u>Original Goals</u>:

- Scalable, Scalapack-compatible "drop-in enhancement"
- Pure MPI-based implementation
- Detailed rewrite based on proven robust/general algorithms

<u>*http://elpa.rzg.mpg.de*</u>

A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H.-J. Bungartz, H. Lederer, The Journal of Physics: Condensed Matter 26, 213201 (2014).

A Massively Parallel Dense Eigensolver: "ELPA"

$$\underline{\underline{h}}\,\underline{c}_k = \epsilon_k\,\underline{\underline{s}}\,\underline{c}_k$$

Given a matrix H and metric S (dimension *N*),
find *M* eigenvalue/eigenvector pairs ε_k/c_k

Original Goals:

- Scalable, Scalapack-compatible "drop-in enhancement"
- Pure MPI-based implementation
- Detailed rewrite based on proven robust/general algorithms

Today:

- Standalone open-source (LGPL) library,
  used in other major codes (cp2k, Quantum Espresso, VASP, ...)
- Optional support for shared-memory systems (OpenMP)

http://elpa.rzg.mpg.de

A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H.-J. Bungartz, H. Lederer,
The Journal of Physics: Condensed Matter 26, 213201 (2014).

Taking Apart the Eigenvalue Problem

$$\underline{\underline{h}}\,\underline{c}_k = \epsilon_k\,\underline{\underline{s}}\,\underline{c}_k$$

Generalized (non-orthogonal) eigenvalue problem:

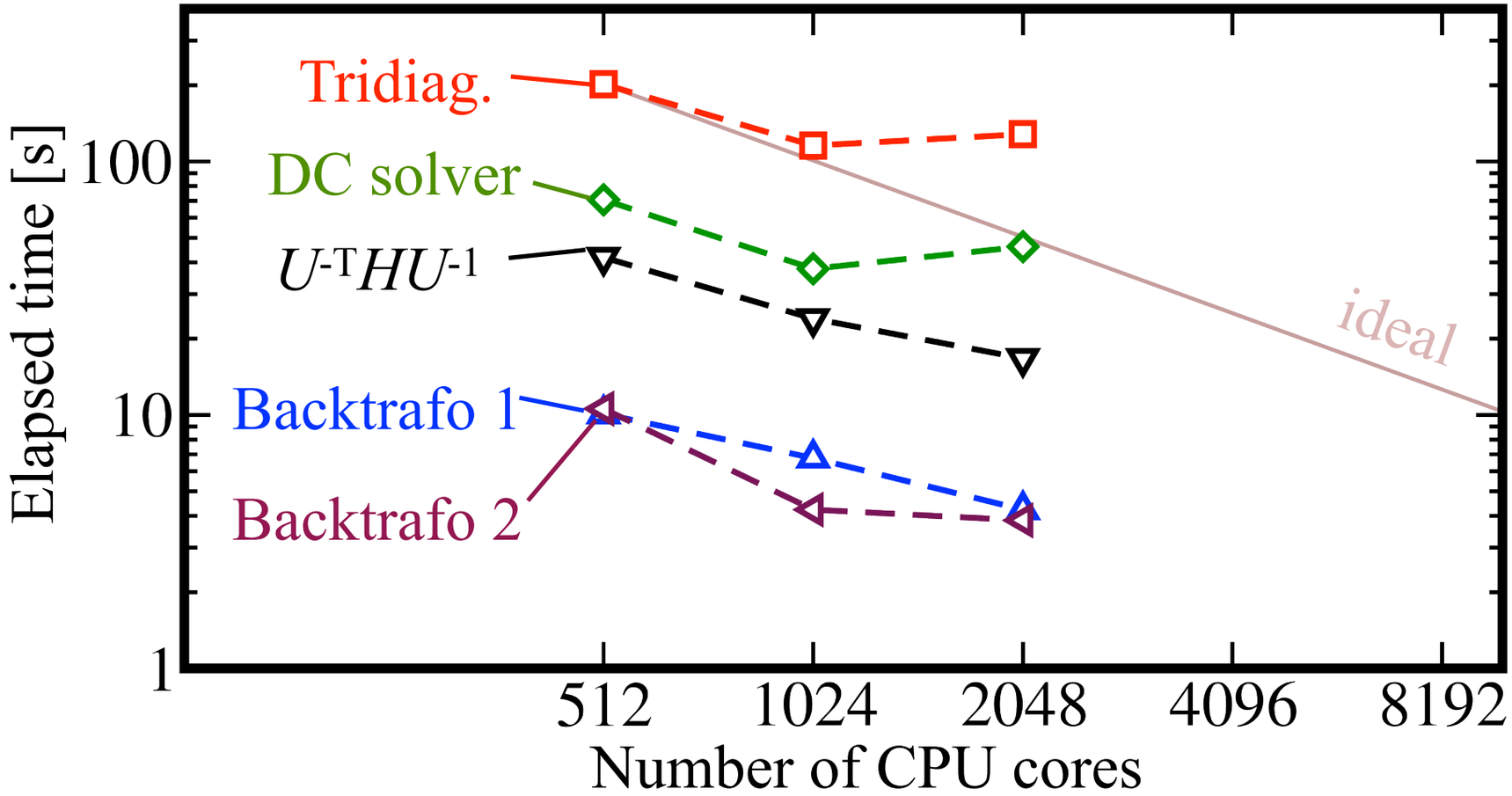
- Transform to orthogonal form: $U^{-T}HU^{-1}$
- Transform orthogonal *H'* to *tridiagonal* form
- Solve *tridiagonal* eigenproblem
- Backtransform (1) solution to standard form
- Backtransform (2) standard to general form

Taking Apart the Eigenvalue Problem

$$\underline{\underline{h}}\,\underline{c}_k = \epsilon_k\,\underline{\underline{s}}\,\underline{c}_k$$

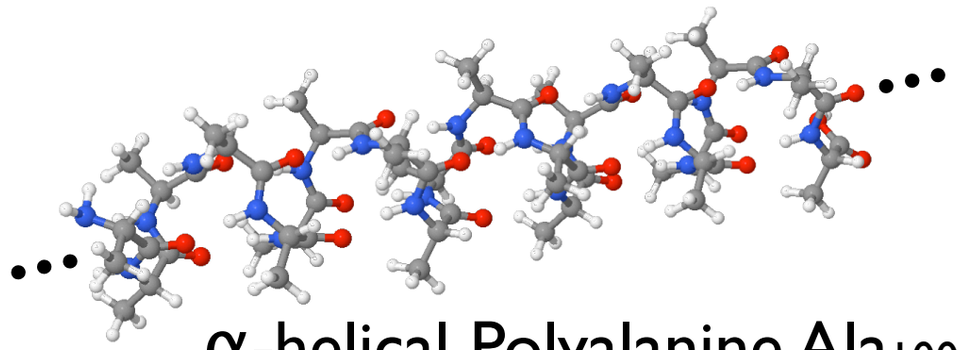
Generalized (non-orthogonal) eigenvalue problem:

- Transform to orthogonal form: $U^{-T}HU^{-1}$
- Transform orthogonal *H'* to *tridiagonal* form
- Solve *tridiagonal* eigenproblem
- Backtransform (1) solution to standard form
- Backtransform (2) standard to general form

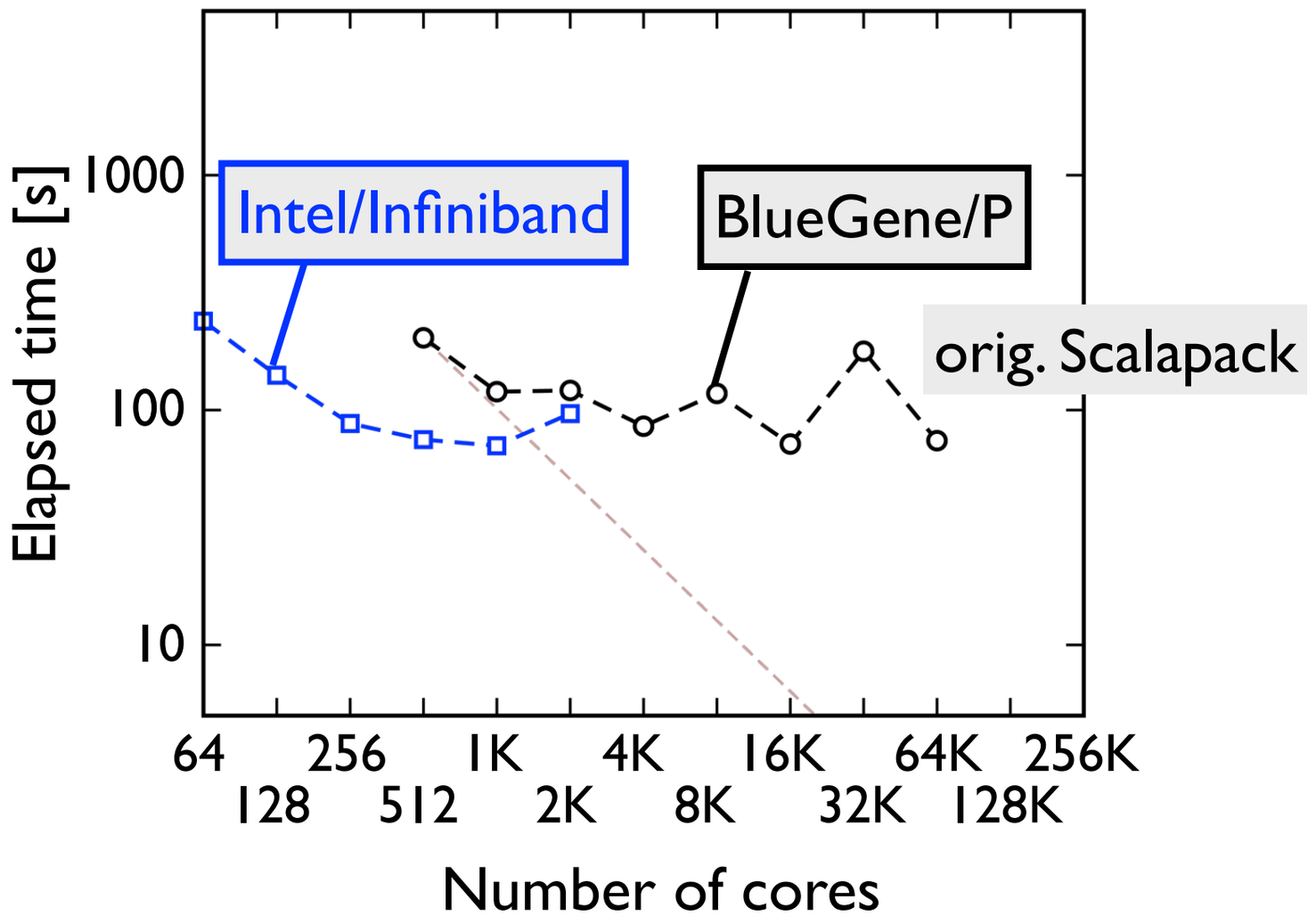


α-helical Polyalanine Ala_{100}, BlueGene/P

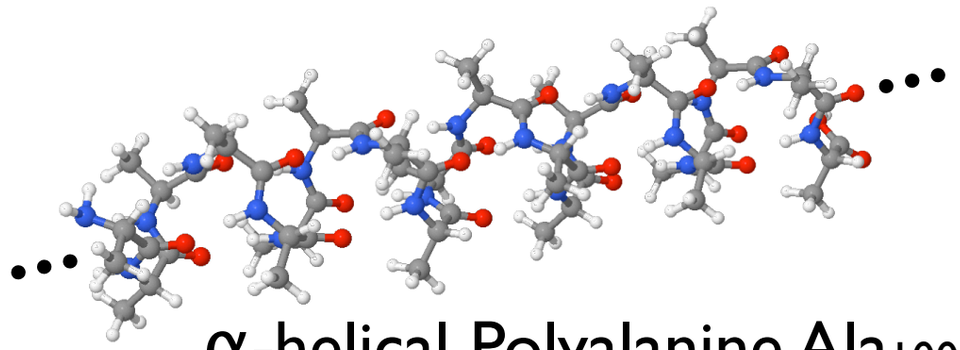
A Careful Rewrite Can Improve Scaling ("ELPA 1")



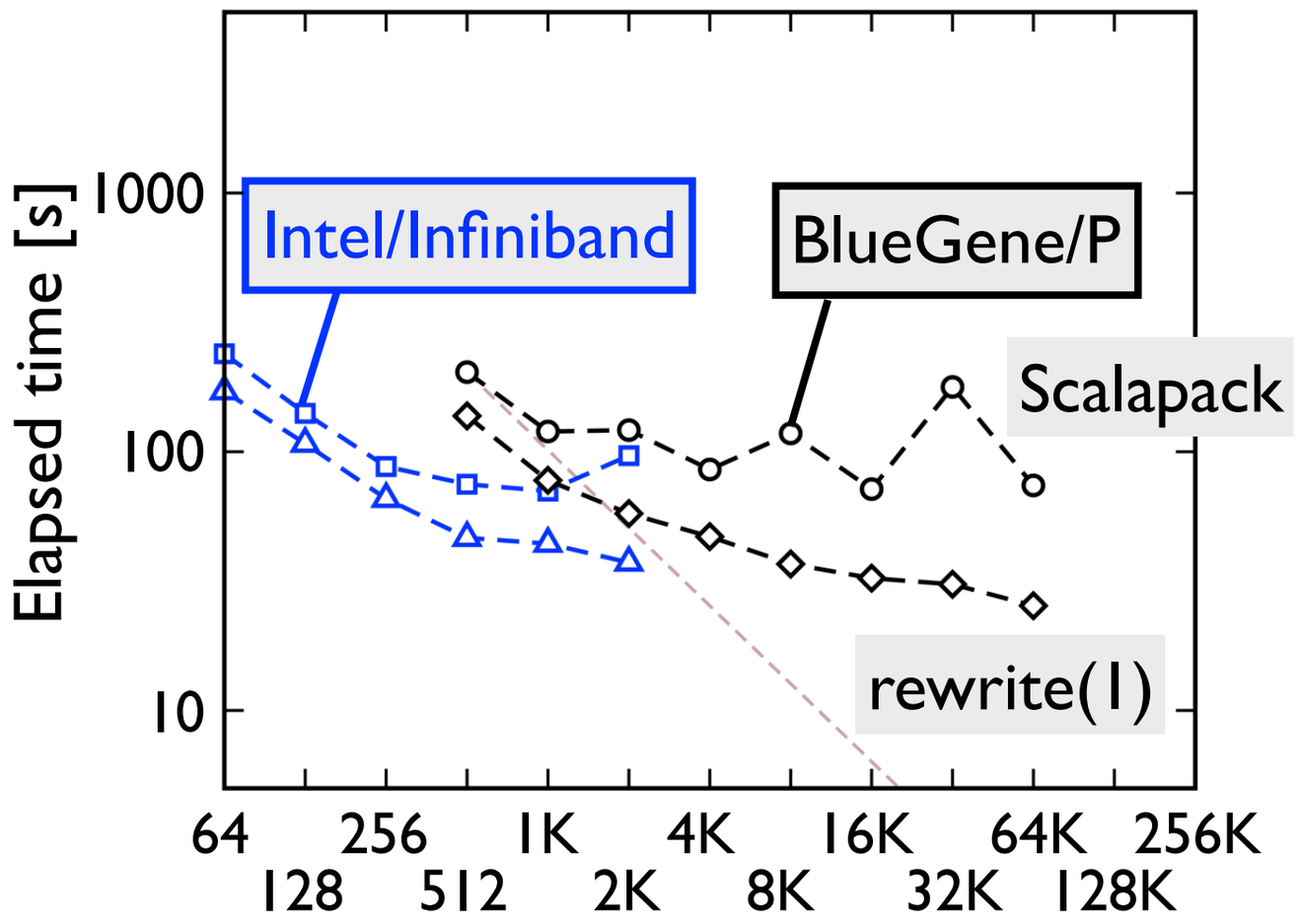
α-helical Polyalanine Ala_{100}
N=27069, *M*=3410
NAO basis set (FHI-aims)



A Careful Rewrite Can Improve Scaling ("ELPA 1")



α-helical Polyalanine Ala_{100}
N=27069, M=3410
NAO basis set (FHI-aims)

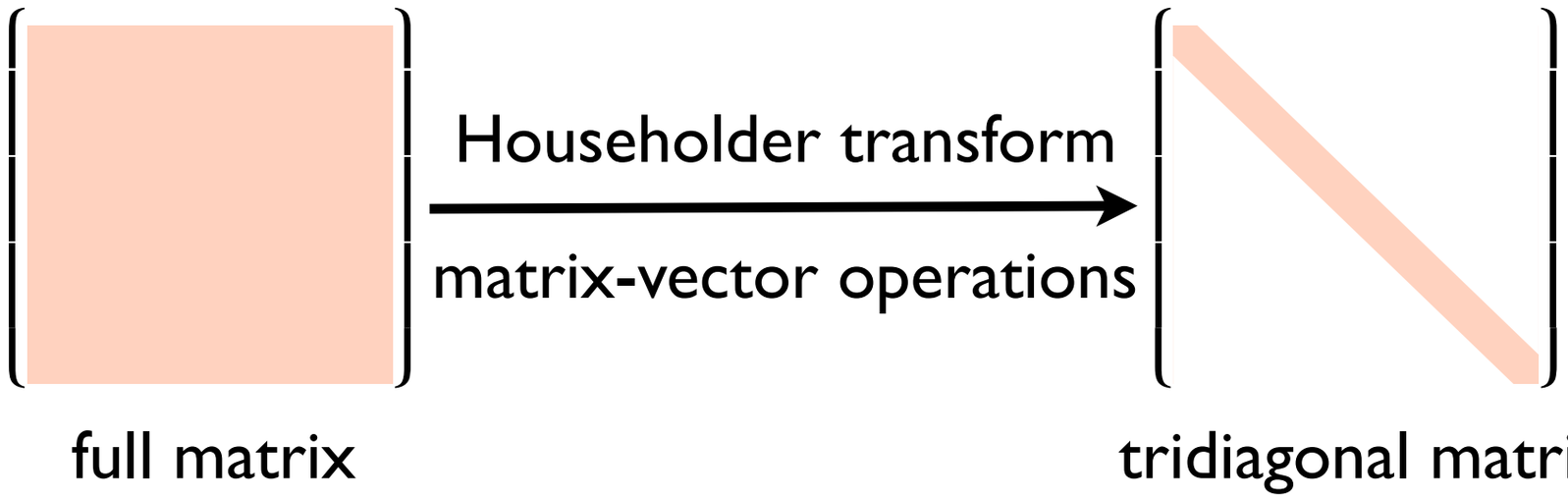


Auckenthaler et al., Parallel Computing (2011)

Algorithmic Improvement: 2-Step Tridiagonalization

Remaining chief bottleneck: Tridiagonalization

"Conventional" reduction:



full matrix

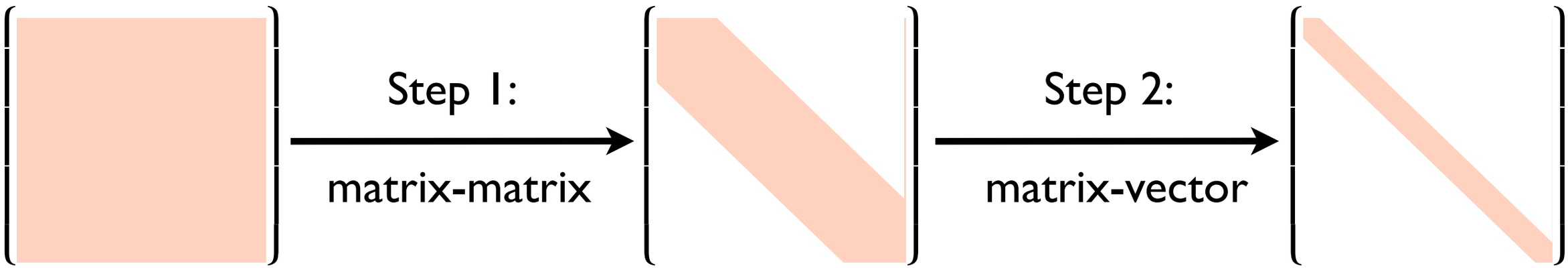
tridiagonal matrix

Algorithmic Improvement: 2-Step Tridiagonalization

Remaining chief bottleneck: Tridiagonalization

"Two-step" reduction:

C. *Bischof, B. Lang, X. Sun, ACM Trans. Math. Software* ***26****, 581 (2000).*



full matrix band matrix tridiagonal matrix

But extra back transform necessary - benefit shrinks for *M* approaching *N*

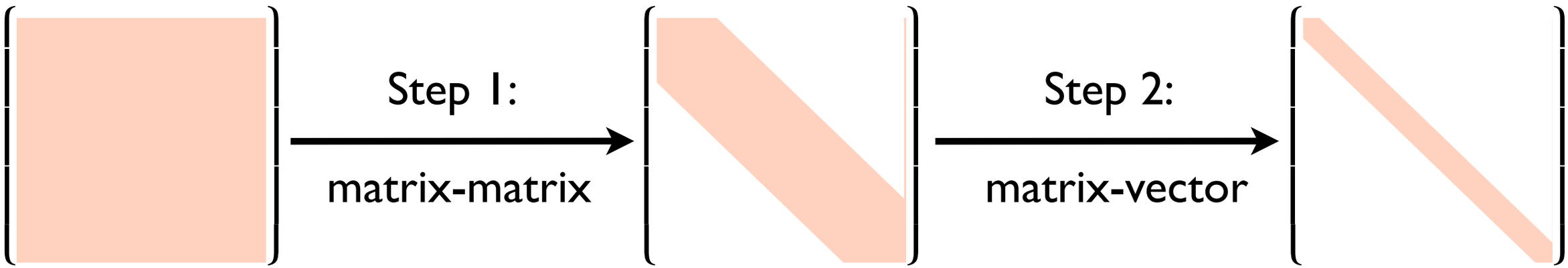
Auckenthaler, Blum, Bungartz, Huckle, Johanni, Krämer, Lang, Lederer, Willems, Parallel Computing (2011)

Algorithmic Improvement: 2-Step Tridiagonalization

Remaining chief bottleneck: Tridiagonalization

"Two-step" reduction:

C. *Bischof, B. Lang, X. Sun, ACM Trans. Math. Software* ***26****, 581 (2000).*



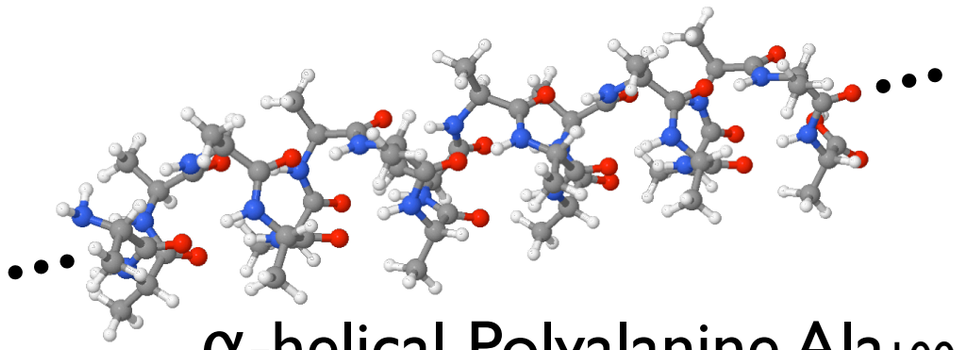
But extra back transform necessary - benefit shrinks for *M* approaching *N*

Massively parallel two-step tridiagonalization:

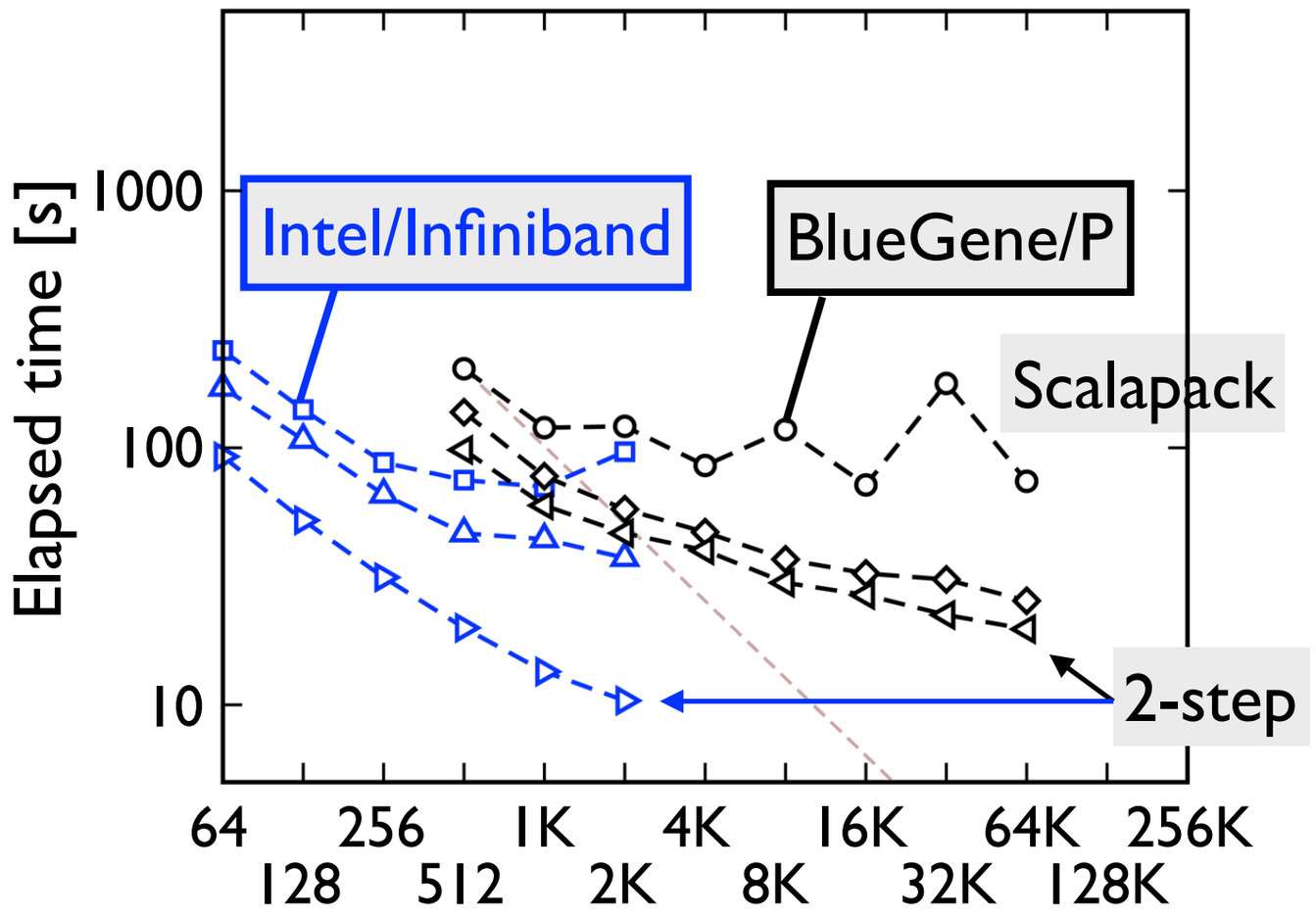
- 2-dimensional data layout for eigenvectors
- Heavily optimized backtransform steps for eigenvectors (adaptive data layout, architecture-specific linear algebra kernels - cache blocking)

Auckenthaler, Blum, Bungartz, Huckle, Johanni, Krämer, Lang, Lederer, Willems, Parallel Computing (2011)

ELPA, Two-Step Solver



α-helical Polyalanine Ala_{100}
N=27069, *M*=3410
NAO basis set (FHI-aims)



Auckenthaler et al., Parallel Computing (2011)

ELPA Today

Actively developed high-performance library (Kus, Marek, others):

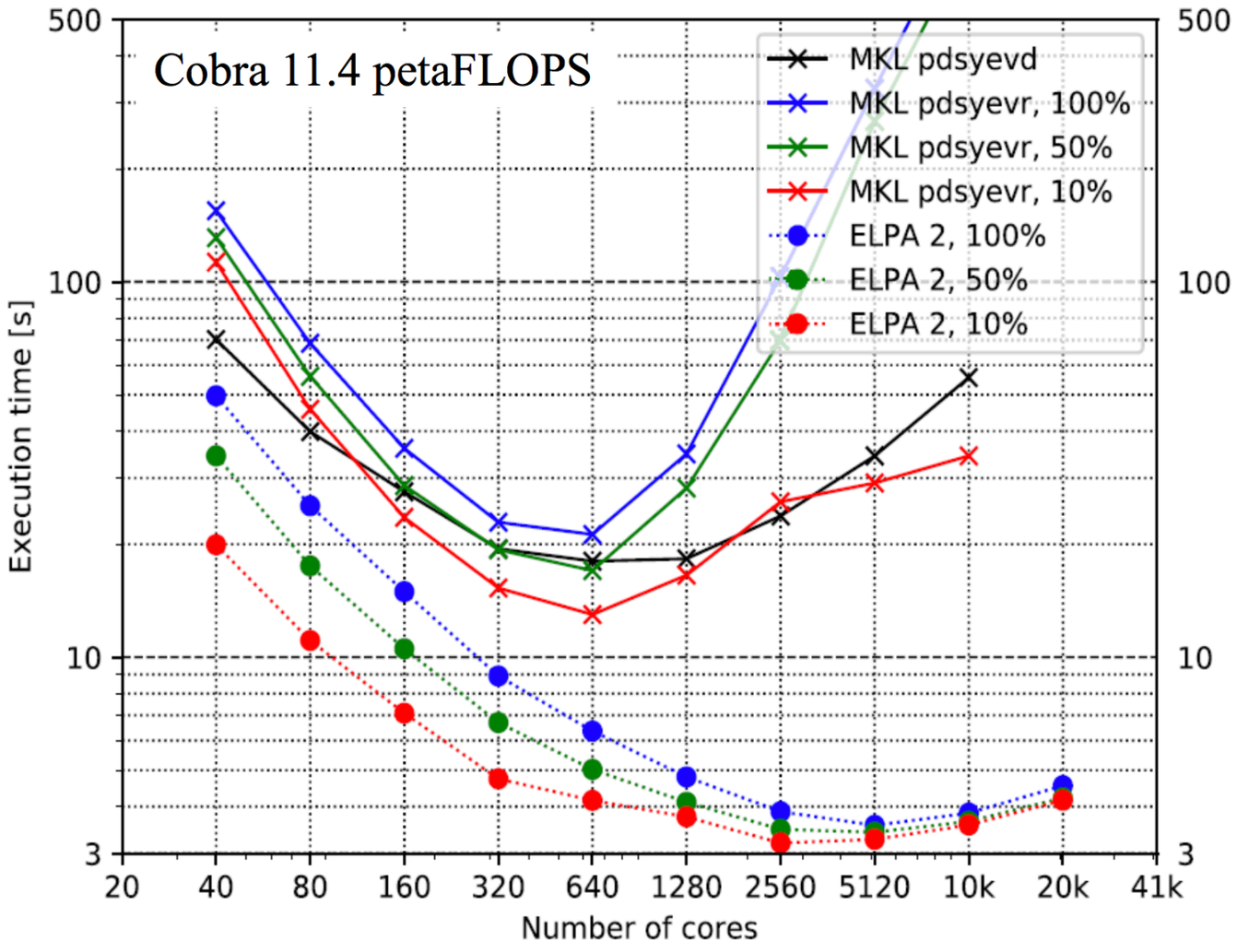


Fig. 3, Kus et al., Comput. Phys. Commun. 2019
Matrix size 20,000, Intel MKL version 2018

ELPA dense eigensolver
http://elpa.mpcdf.mpg.de

- High-performance, massively parallel two-stage tri-diagonalization
- Scales to over 10k CPU cores
- Optimized for present-day supercomputing architectures
- Drop-in enhancement to ScaLAPACK standard and generalized eigensolvers

Auckenthaler et al., Parallel Comput. 2011
Marek et al., J. Phys. Condens. Matter 2014
Kus et al., Comput. Phys. Commun. 2019

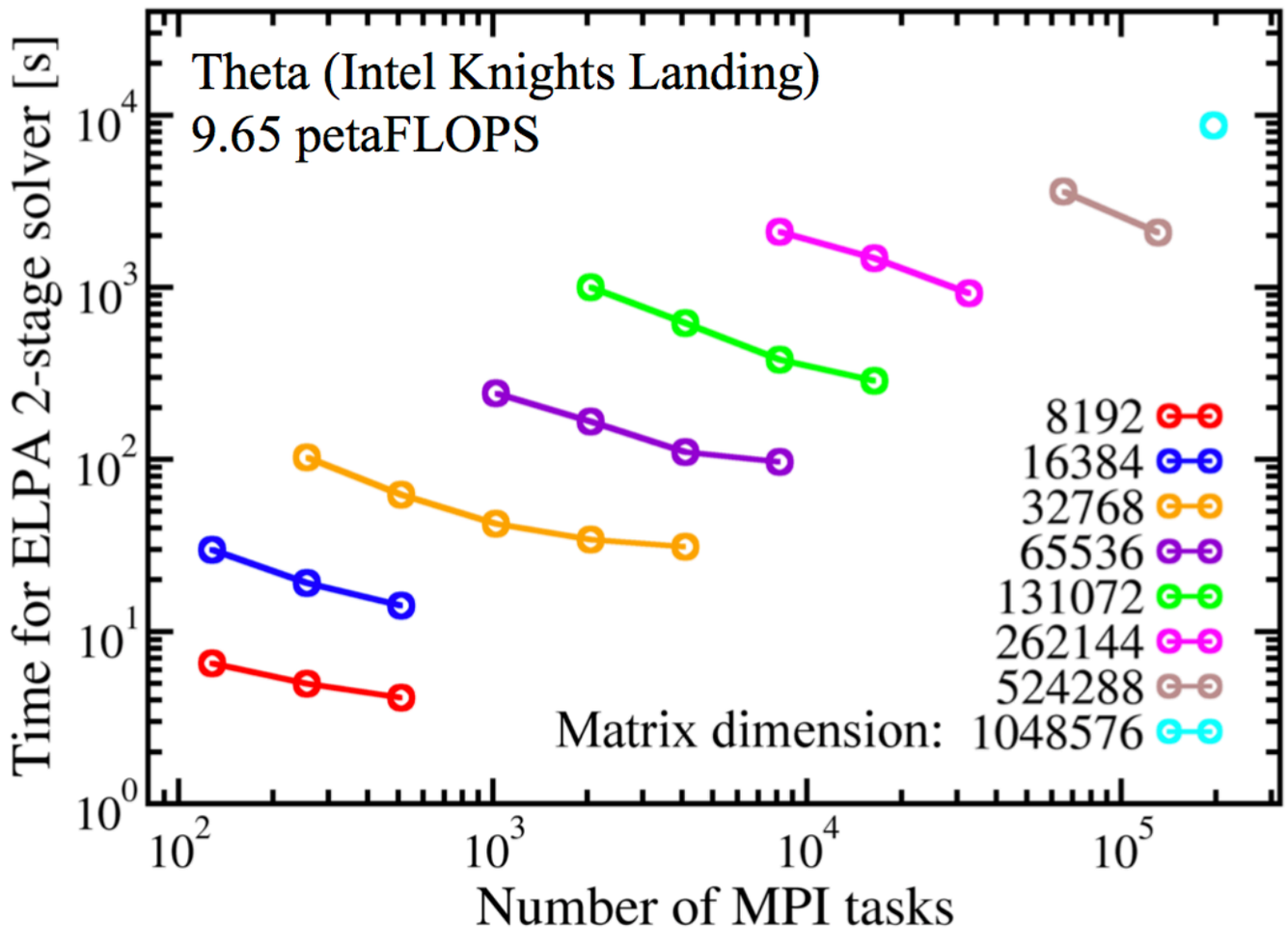
But How to Tackle Larger System Sizes?

Plan A: Have Big Computer, Push the Eigensolver

Benchmark: Alvaro Vazquez-Mayagoitia, ANL

ELPA Eigenvalue Solver

- Efficient full →band →tridiagonal reduction & backtransform
- Dense linear algebra up to full spectrum



Auckenthaler, Blum, Bungartz, Huckle, Johanni, Krämer, Lang, Lederer, Willems, Parallel Computing 37, 783 (2011)

A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H.-J. Bungartz, H. Lederer, The Journal of Physics: Condensed Matter 26, 213201 (2014).

Plan B: Density Matrix Based Approaches

Straightforward Solution: Eigenvalue Solver, $O(N^3)$

$$\psi_k(r) = \sum_i c_{ik}\varphi_i(r) \quad \rightarrow \quad \underline{\underline{h}}\,\underline{c}_k = \epsilon_k \underline{\underline{s}}\,\underline{c}_k \quad \rightarrow \quad n(r) = \sum_k f_k |\psi_k(r)|^2$$

Plan B: Density Matrix Based Approaches

Straightforward Solution: Eigenvalue Solver, $O(N^3)$

$$\psi_k(r) = \sum_i c_{ik}\varphi_i(r) \quad \rightarrow \quad \underline{\underline{h}}\,\underline{c}_k = \epsilon_k \underline{\underline{s}}\,\underline{c}_k \quad \rightarrow \quad n(r) = \sum_k f_k |\psi_k(r)|^2$$

However, in DFT we need $n(r)$ - eigenvectors are not needed:

$$n(r) = \sum_k f_k \sum_{ij} c_{ik}c_{jk}\varphi_i(r)\varphi_j(r) = \sum_{ij} \varphi_i(r) n_{ij} \varphi_j(r)$$

$$\boxed{n_{ij} = \sum_k f_k c_{ik} c_{jk}}$$

Density matrix - sparse!
Decays with distance between basis functions *i,j*

→ Construct density matrix in less than $O(N^3)$ for given h, s?

Many Solution Strategies to Kohn-Sham Problem

$$\underline{\underline{h}}\,\underline{c}_k = \epsilon_k\,\underline{\underline{s}}\,\underline{c}_k$$

Different use cases (basis sets, physics), different "solvers".
Solve (eigenvectors, $O(N^3)$) or circumvent (density matrix)?

Many Solution Strategies to Kohn-Sham Problem

$$\underline{\underline{h}}\,\underline{c}_k = \epsilon_k\,\underline{\underline{s}}\,\underline{c}_k$$

Different use cases (basis sets, physics), different "solvers".
Solve (eigenvectors, $O(N^3)$) or circumvent (density matrix)?

Exact solvers	Iterative solvers	DM: O(N) solvers	Other DM-based approaches
Lapack	Davidson	NTPoly	PEXSI
Scalapack	Projected Preconditioned Conjugate Gradient	Various code-internal and/or proprietary implementations	Orbital Minimization Method
ELPA	Chebychev Filtering		FEAST
EigenExa	Slepc-SIPS		…
Magma	…		
…			

Many Solution Strategies to Kohn-Sham Problem

$$\underline{\underline{h}}\,\underline{c}_k = \epsilon_k\,\underline{\underline{s}}\,\underline{c}_k$$

Different use cases (basis sets, physics), different "solvers".
Solve (eigenvectors, $O(N^3)$) or circumvent (density matrix)?

Exact solvers	Iterative solvers	DM: O(N) solvers	Other DM-based approaches
Lapack Scalapack ELPA EigenExa Magma …	Davidson Projected Preconditioned Conjugate Gradient Chebychev Filtering Slepc-SIPS …	NTPoly Various code-internal and/or proprietary implementations	PEXSI Orbital Minimization Method FEAST …
Robust General	(Essentially) robust $N_{basis} >> N_{ev}$	Sparse H, S Nonmetallic systems	Sparse H, S can depend on XC

Many Solution Strategies to Kohn-Sham Problem

$$\underline{\underline{h}}\,\underline{c}_k = \epsilon_k\,\underline{\underline{s}}\,\underline{c}_k$$

Different use cases (basis sets, physics), different "solvers".
Solve (eigenvectors, $O(N^3)$) or circumvent (density matrix)?

Exact solvers	Iterative solvers	DM: O(N) solvers	Other DM-based approaches
Lapack Scalapack ELPA EigenExa Magma …	Davidson Projected Preconditioned Conjugate Gradient Chebychev Filtering Slepc-SIPS …	NTPoly Various code-internal and/or proprietary implementations	PEXSI Orbital Minimization Method FEAST …
Robust General	(Essentially) robust $N_{basis} >> N_{ev}$	Sparse H, S Nonmetallic systems	Sparse H, S can depend on XC

Pole Expansion and Selected Inversion (PEXSI)

Expand density matrix $\mathbf{P}$ in terms of rational functions (poles)

$$\mathbf{P} = \sum_l \mathrm{Im}\left(\frac{w_l}{\mathbf{H} - (z_l + \mu)\mathbf{S}}\right)$$

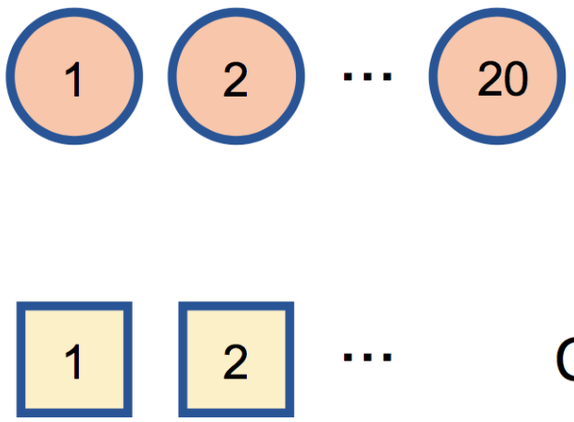
$\mathbf{P}$: Density matrix
$\mathbf{H}$: Hamiltonian matrix
$\mathbf{S}$: Overlap matrix
z_l: Shift (pole)
w_l: Weight
μ: Chemical potential

- Computational steps:
  1) Determine pole expansion parameters $\{z_l, w_l\}$
  2) Compute all poles in parallel by process groups
  3) Sum over poles to get density matrix and other quantities

Lin et al., Commun. Math. Sci. 2009
Lin et al., J. Phys. Condens. Matter 2013
Lin et al., J. Phys. Condens. Matter 2014
Jia and Lin, J. Chem. Phys. 2017

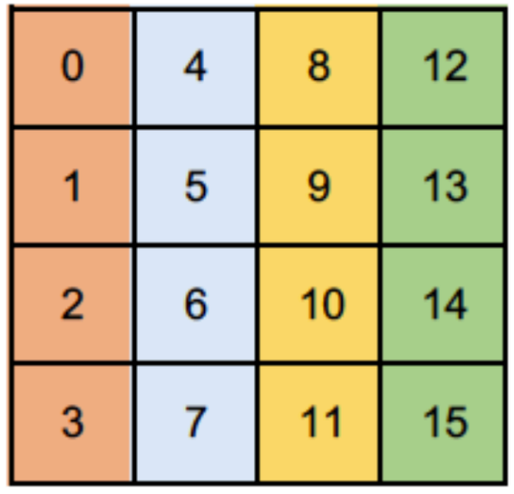
- Applicable to insulators, semiconductors as well as metals
- Selected inversion: Evaluate needed elements of $(\mathbf{H} - (z_l - \mu)\mathbf{S})^{-1}$

Pole Expansion and Selected Inversion (PEXSI)



Lv. 1
Pole expansion
(20~100 poles)

Lv. 2
Chemical potential search
(a few trial μ points)



Lv. 3
Parallel selected inversion
(1,000+ CPU cores)

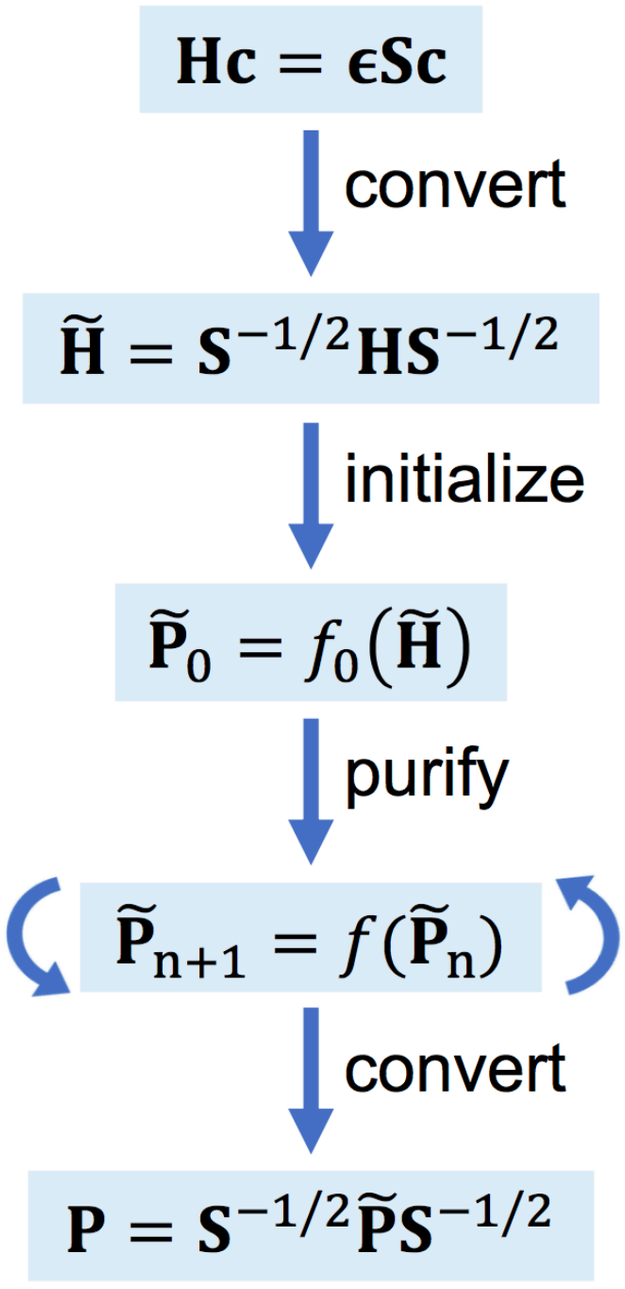
PEXSI sparse density matrix solver
http://www.pexsi.org

$$\mathbf{P} = \sum_l \mathrm{Im}\left(\frac{w_l}{\mathbf{H} - (z_l + \mu)\mathbf{S}}\right)$$

- Computational complexity (semi-local XC)
  - 1D system: $O(N)$
  - 2D system: $O(N^{1.5})$
  - 3D system: $O(N^2)$
- Scales to over 100k CPU cores

Lin et al., Commun. Math. Sci. 2009
Lin et al., J. Phys. Condens. Matter 2013
Lin et al., J. Phys. Condens. Matter 2014
Jia and Lin, J. Chem. Phys. 2017

NTPoly: *O*(*N*) Approaches



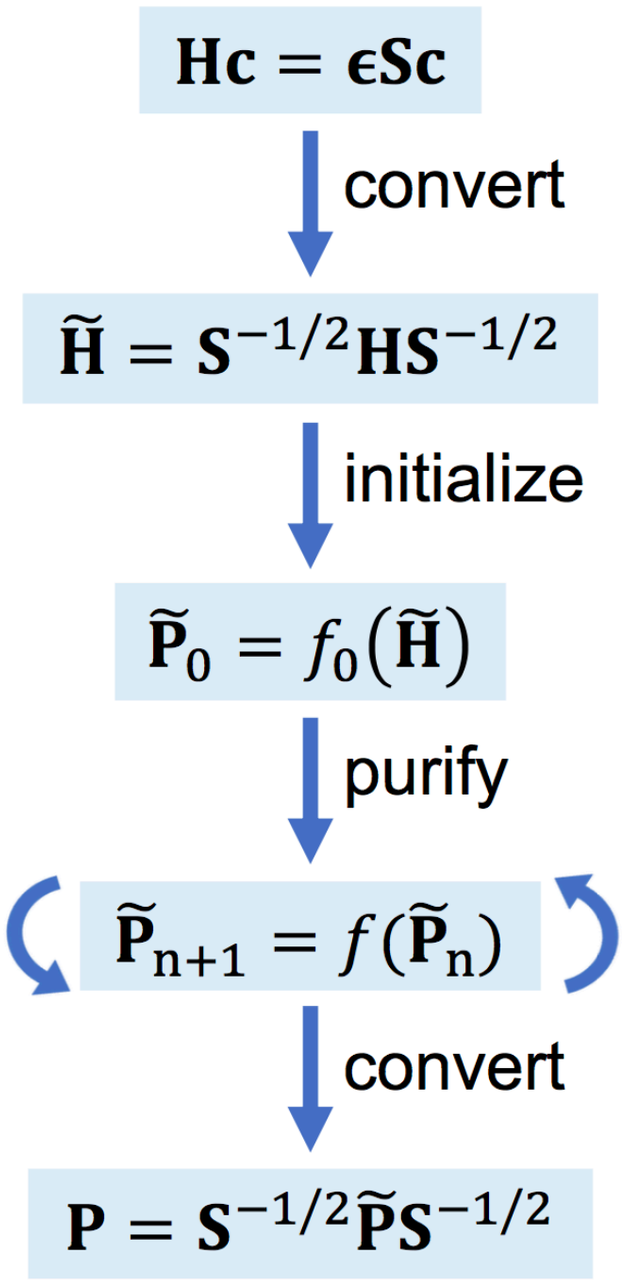
- Zero-temperature density matrix $\mathbf{P}$ must be: (assuming orthogonal basis)
  - Hermitian: $\mathbf{P} = \mathbf{P}^*$
  - Normalized: $\mathrm{Tr}(\mathbf{P}) = \mathrm{N}_{\mathrm{electron}}$
  - Idempotent: $\mathbf{P} = \mathbf{P}^2$

- Computational steps:
  1) Convert to orthogonal basis
  2) Construct initial guess of density matrix
  3) Iteratively purify density matrix to satisfy the above three conditions
  4) Convert back to non-orthogonal basis

- Established methods for linear scaling electronic structure theory: One million atoms possible

Goedecker, Rev. Mod. Phys. 1999
Bowler and Miyazaki, Rep. Prog. Phys. 2012

NTPoly: *O*(*N*) Approaches



- Purification function $f(\widetilde{\mathbf{P}}_n)$ is often a matrix polynomial of order m
  - Canonical purification (m = 3)
  - Trace resetting purification (m = 2, 3, 4, ...)
  - Generalized canonical purification (m = 3)

- Example: 2nd order trace resetting purification

$$\sigma_n = \mathrm{sign}(N_{electron} - \mathrm{Tr}(\mathbf{P}_n))$$
$$\mathbf{P}_{n+1} = \mathbf{P}_n + \sigma_n(\mathrm{I} - \mathbf{P}_n)\mathbf{P}_n$$

- Sparse matrix algebra essential for linear scaling

Goedecker, Rev. Mod. Phys. 1999
Bowler and Miyazaki, Rep. Prog. Phys. 2012
Palser and Manolopoulos, Phys. Rev. B 1998
Niklasson, Phys. Rev. B 2002
Truflandier et al., J. Chem. Phys. 2016

Many Different Solvers - How to Unify Access?

Electronic structure codes

?

Solvers

ELPA | PEXSI

SLEPc-SIPs | libOMM

NTPoly | Many more

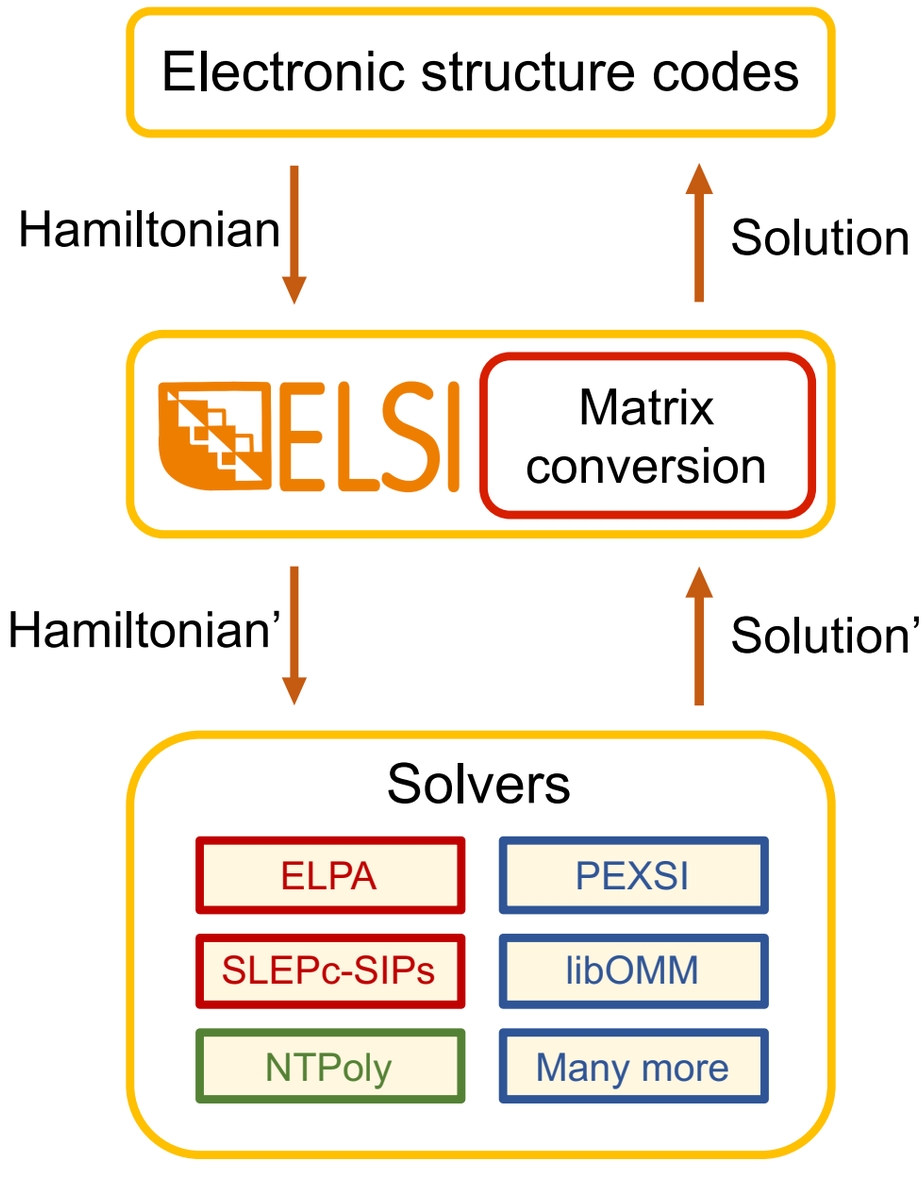
? Replicated infrastructure to implement solvers efficiently

? Conversion between a variety of matrix storage formats

? Complexity in solver selection for different problems

ELSI: Connecting Electronic Structure Codes and Solvers

Yu et al., Comput. Phys. Commun. 2018
http://elsi-interchange.org
http://git.elsi-interchange.org/elsi-devel/elsi-interface

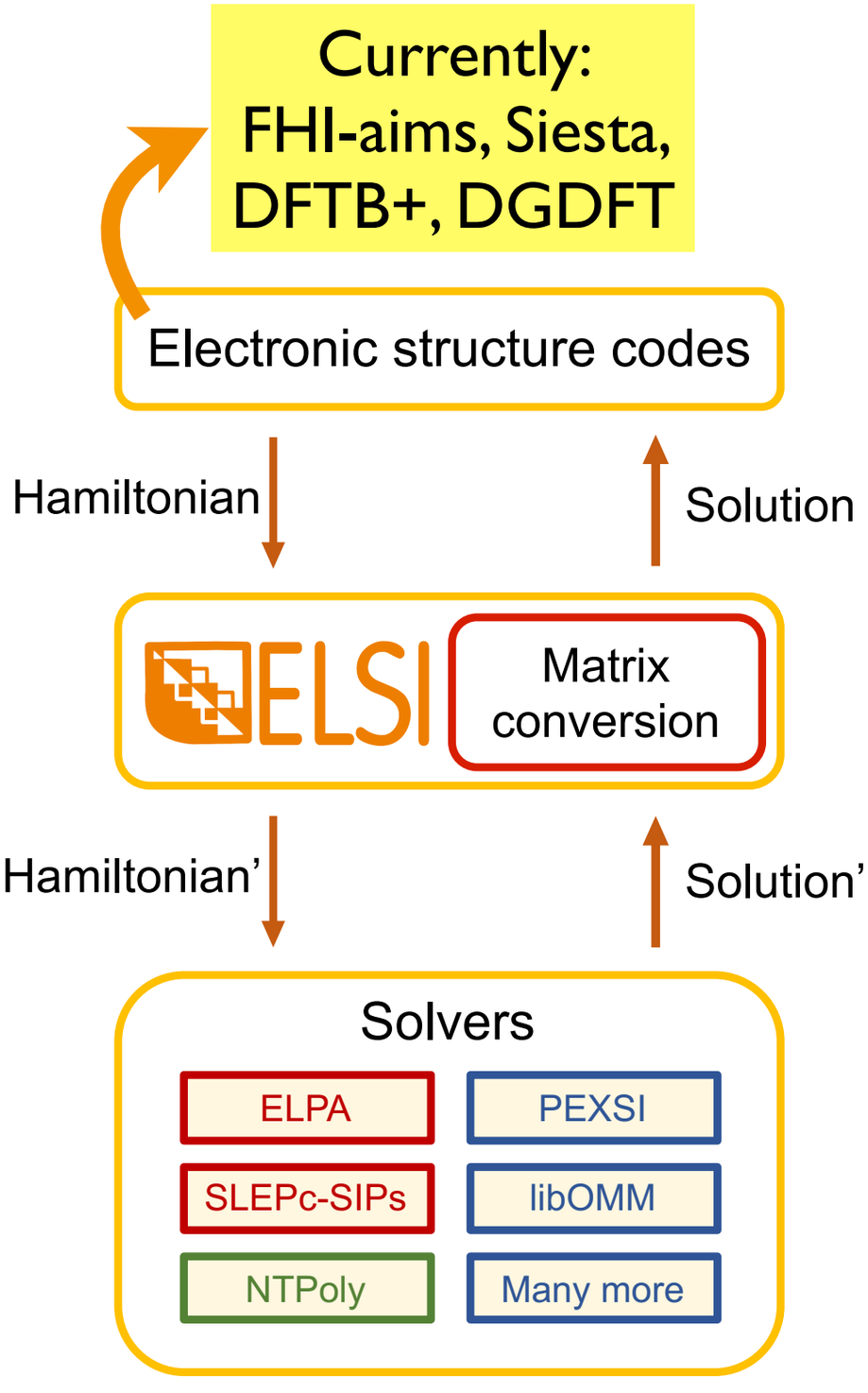


Unified interface connecting KS-DFT codes and solvers

Fast, automatic matrix format conversion and redistribution

Recommendation of optimal solver based on benchmarks

ELSI: Connecting Electronic Structure Codes and Solvers



Yu et al., Comput. Phys. Commun. 2018
http://elsi-interchange.org
http://git.elsi-interchange.org/elsi-devel/elsi-interface

Unified interface connecting KS-DFT codes and solvers

Fast, automatic matrix format conversion and redistribution

Recommendation of optimal solver based on benchmarks

Portability, Extendability, Sustainability

- Designed for rapid integration into a variety of electronic structure codes
- Compatible with common workflows
  - Single self-consistent field (SCF)
  - Multiple SCF cycles (geometry relaxation or molecular dynamics)
- Supports density matrix solvers and eigensolvers on equal footing
- Technical settings adjustable for experienced users
- Object-oriented: Concurrent instances

ELSI API

```
elsi_init
elsi_set_parameters
while (geometry not converged) do
  while (SCF not converged) do
    elsi_{ev|dm}
  end while
  elsi_reinit
end while
elsi_finalize
```

Portability, Extendability, Sustainability

Victor Yu

Git commit to add NTPoly support into FHI-aims: 10 lines of code

```
--- a/read_control.f90
+++ b/read_control.f90
@@ -12581,6 +12581,12 @@ subroutine read_control
+  case("elsi_ntpoly_method")
+    read(inputline,*,end=88,err=99) desc_str,elsi_ntpoly_method
+  case("elsi_ntpoly_tol")
+    read(inputline,*,end=88,err=99) desc_str,elsi_ntpoly_tol
+  case("elsi_ntpoly_filter")
+    read(inputline,*,end=88,err=99) desc_str,elsi_ntpoly_filter

--- a/elsi_wrapper.f90
+++ b/elsi_wrapper.f90
@@ -265,6 +265,10 @@ subroutine aims_init_elsi
+  case(SOLVER_NTPOLY)
+    call elsi_set_ntpoly_method(eh,elsi_ntpoly_method)
+    call elsi_set_ntpoly_tol(eh,elsi_ntpoly_tol)
+    call elsi_set_ntpoly_filter(eh,elsi_ntpoly_filter)
```

ELSI Functionality

Functionalities

- Eigensolvers: ELPA, EigenExa, SLEPc, LAPACK
- Density matrix solvers: libOMM, PEXSI, NTPoly
- Parallel solution for spin-polarized and periodic systems
- Dense and sparse matrix formats, arbitrary distribution
- Parallel matrix format conversion and I/O

Code freely available at:
http://elsi-interchange.org

Portability

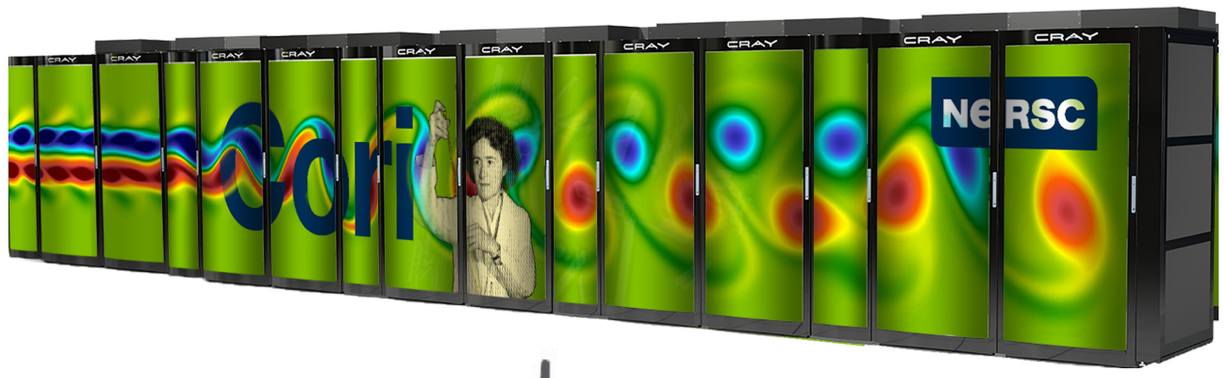
- CMake build system supports Cray, GNU, IBM, Intel, PGI compilers
- From laptops to supercomputers (Cobra, Cori, Mira, Sierra, Theta, …)
- Provides Fortran, C, C++ programming interfaces
- Part of CECAM Electronic Structure Library (ESL): Distribution of shared open-source libraries in the electronic structure community
- http://gitlab.com/ElectronicStructureLibrary/esl-bundle

Performance: Solver Benchmarks on Equal Footing

Victor Yu

http://www.nersc.gov/edison

http://www.nersc.gov/cori



[Edison Cray XC30]

Processor: Intel Ivy Bridge
Interconnect: Cray Aries

5,586 compute nodes
134,064 processing cores
2.57 Petaflops

[Cori-Haswell Cray XC40]

Processor: Intel Haswell
Interconnect: Cray Aries

2,388 compute nodes
76,416 processing cores
2.81 Petaflops

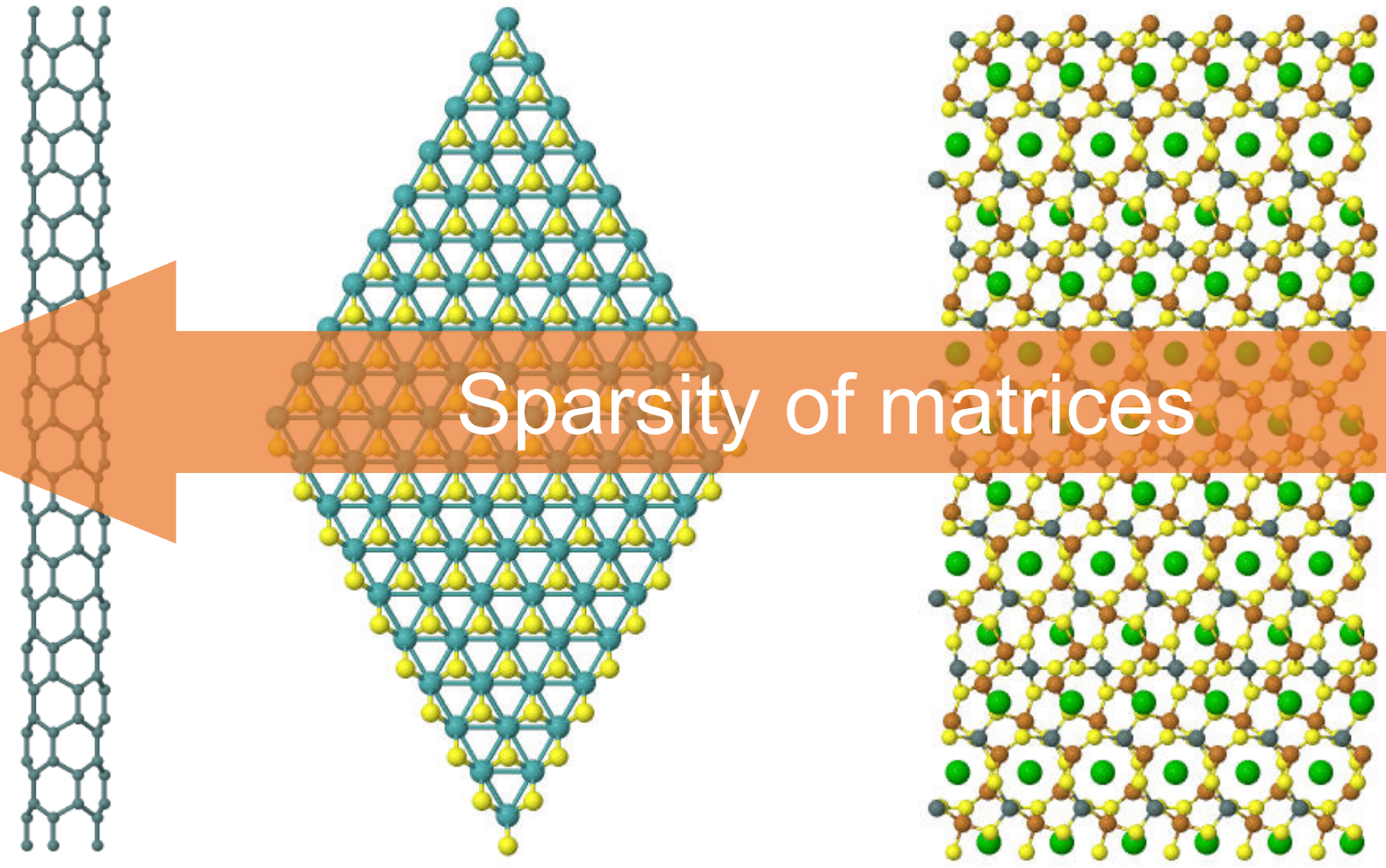
[Cori-KNL Cray XC40]

Processor: Intel Knights Landing
Interconnect: Cray Aries

9,688 compute nodes
658,784 processing cores
29.5 Petaflops

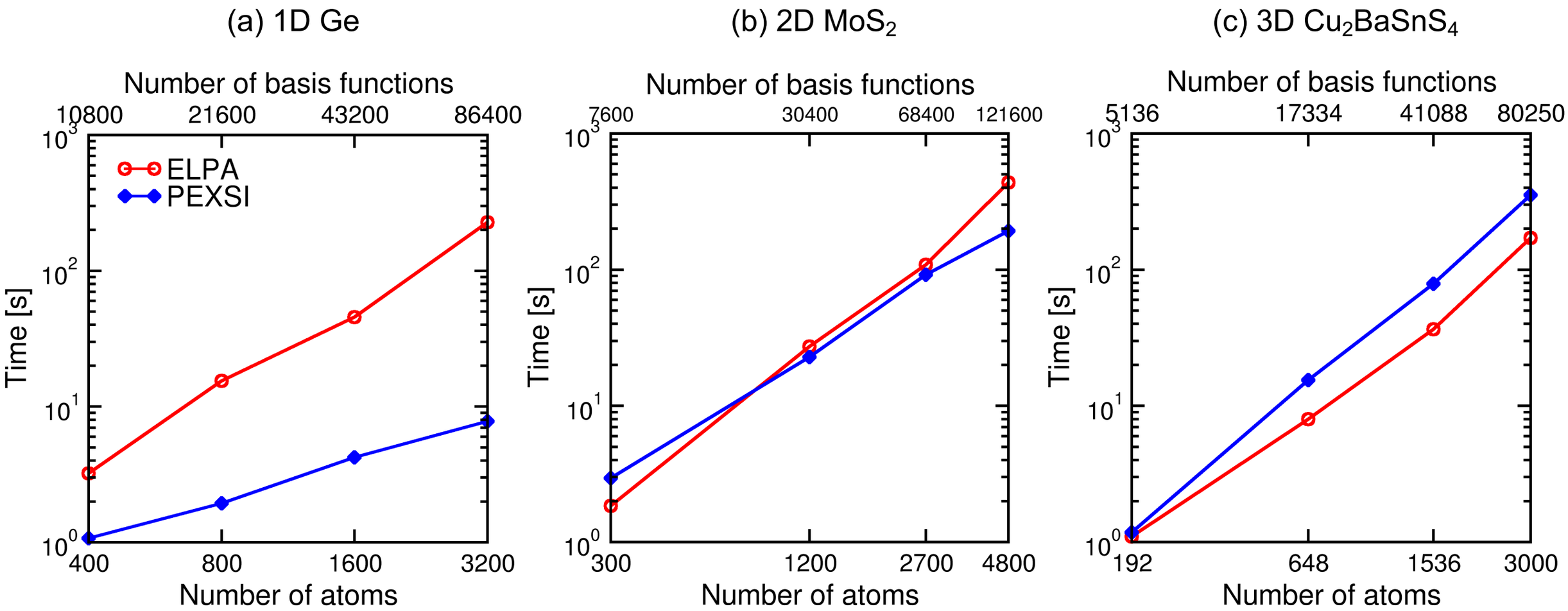
Performance: Solver Benchmarks on Equal Footing

1D	2D	3D
(a) Ge	(b) MoS_2	(c) Cu_2BaSnS_4



Sparsity of matrices

Example: FHI-aims Basis Sets - ELPA vs. PEXSI



- DFT-PBE
- FHI-aims (all-electron)
- 2,560 CPU cores on Cori-Haswell

PEXSI faster for large low-dimensional (sparse) systems

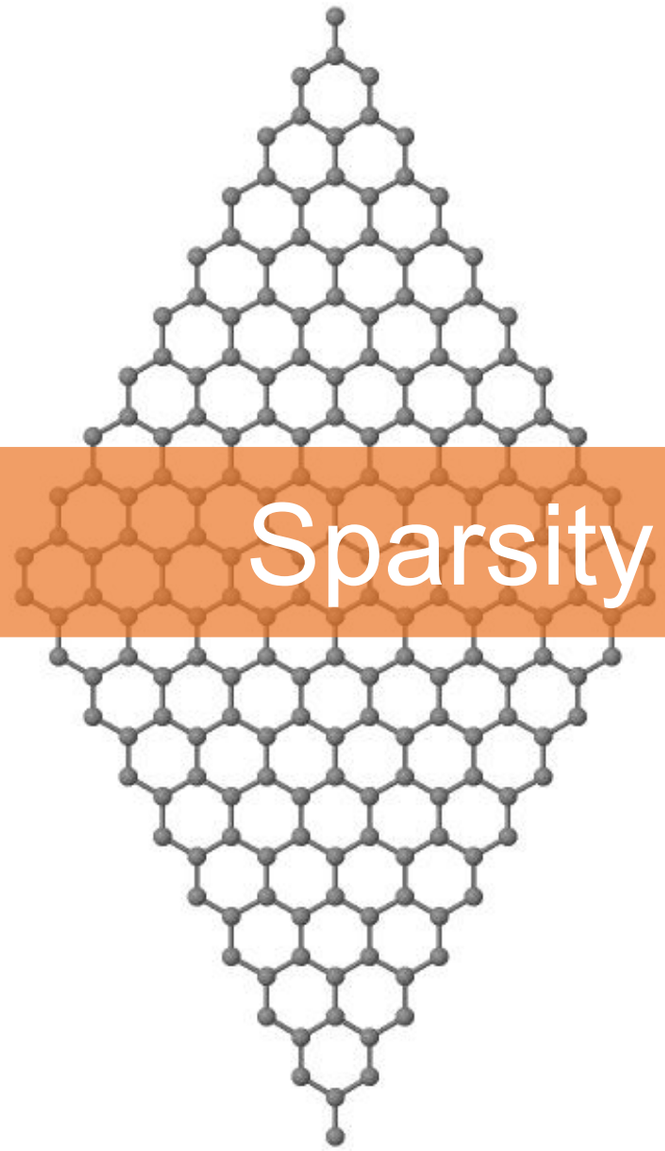
PEXSI: Semilocal DFT, $O(N)$ - $O(N^2)$ for large systems

Lin et al., Commun. Math. Sci. 7, 755 (2009); Lin et al., J. Phys.: Condens. Matter 25, 295501 (2013); Lin et al., J. Phys: Condens. Matter 26, 305503 (2014)

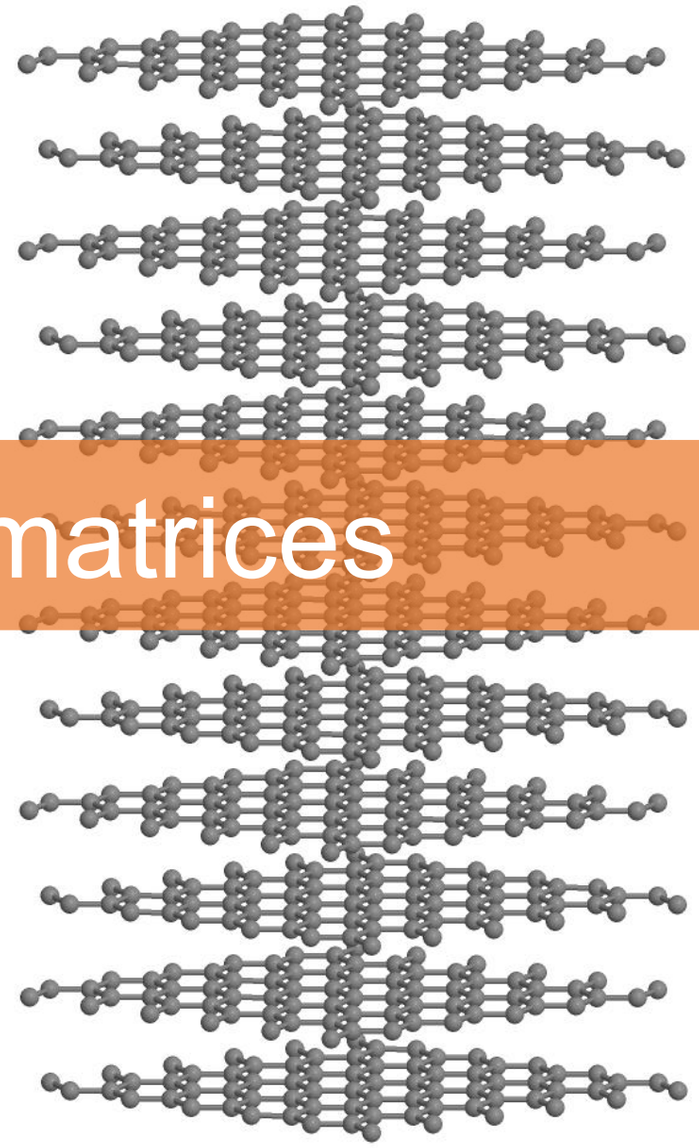
Performance: Solver Benchmarks on Equal Footing

1D
(a) CNT

2D
(b) Graphene

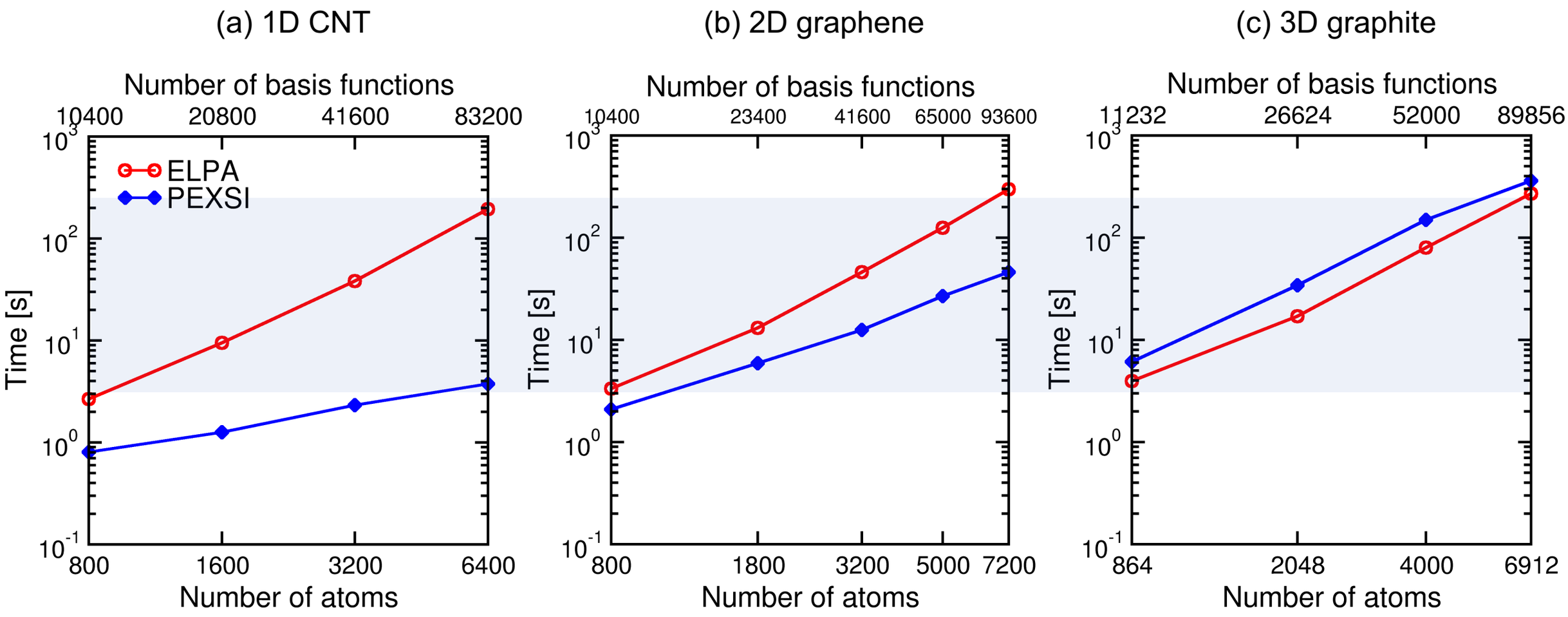


3D
(c) Graphite



Sparsity of matrices

Example: Siesta Basis Sets - ELPA vs. PEXSI



- DFT-PBE
- SIESTA (pseudopotential)
- 1,920 CPU cores on Edison

PEXSI faster for large low-dimensional (sparse) systems

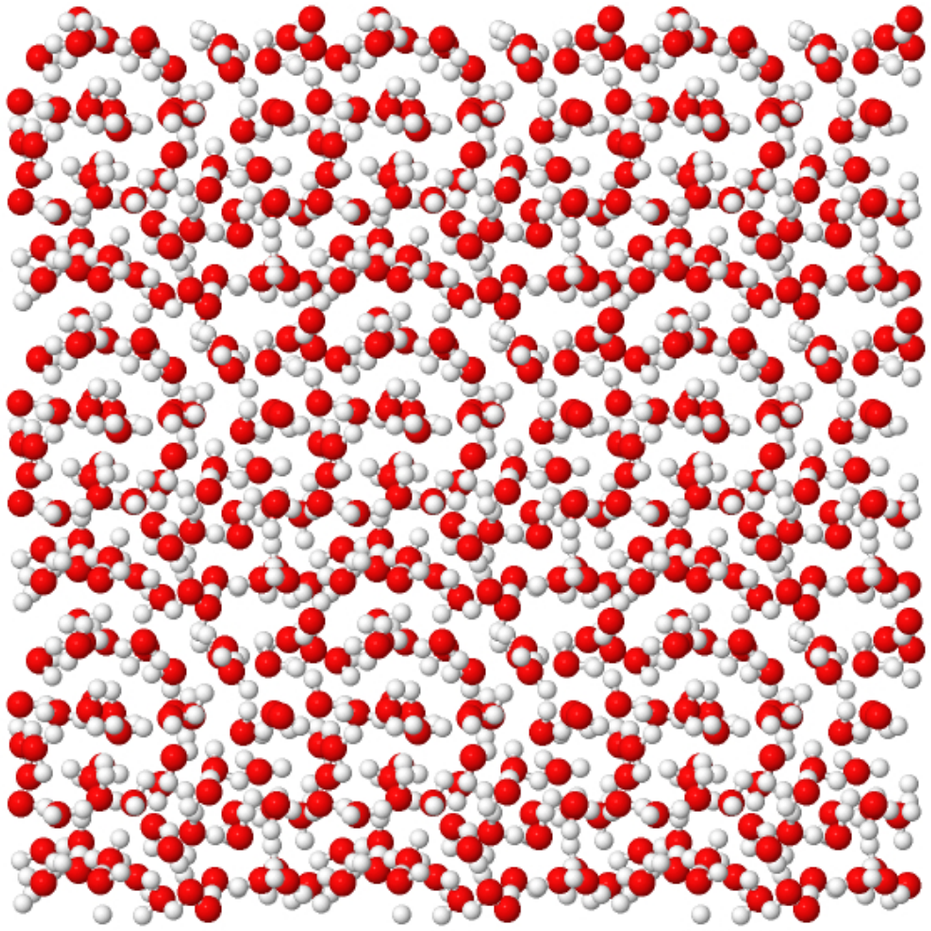
PEXSI: Semilocal DFT, $O(N)$ - $O(N^2)$ for large systems

Lin et al., Commun. Math. Sci. 7, 755 (2009); Lin et al., J. Phys.: Condens. Matter 25, 295501 (2013); Lin et al., J. Phys: Condens. Matter 26, 305503 (2014)

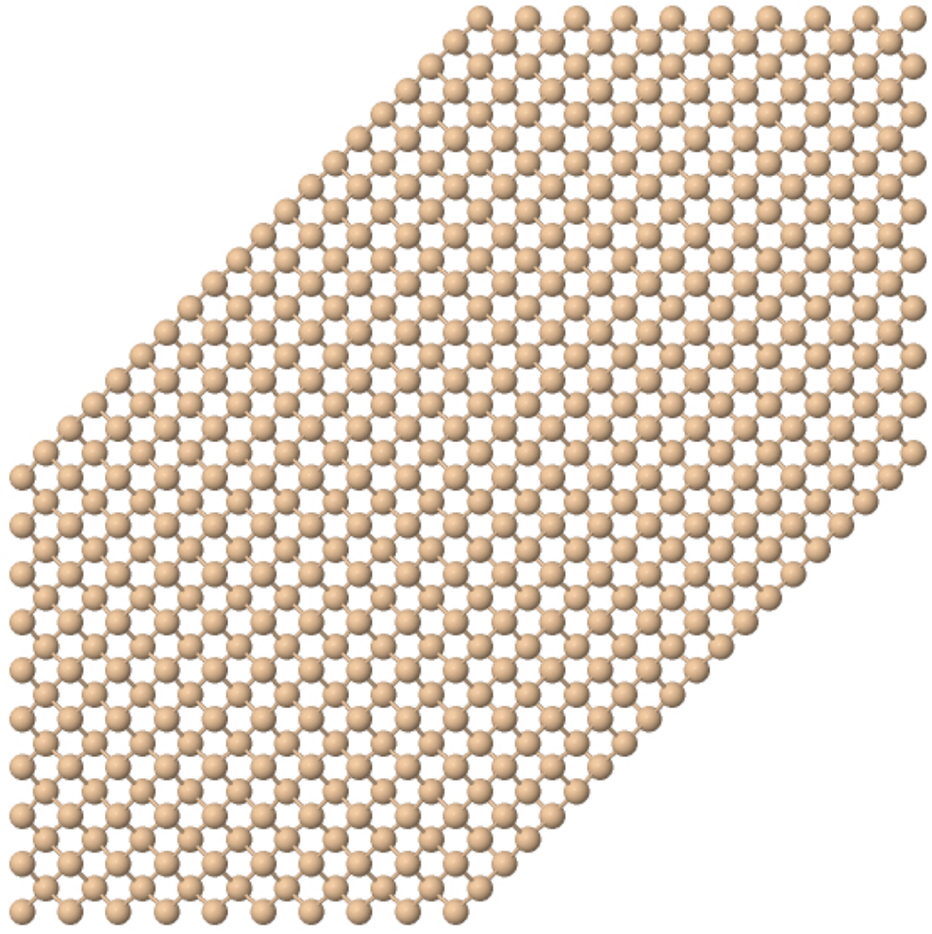
Performance: Solver Benchmarks on Equal Footing

3D Periodic Systems:

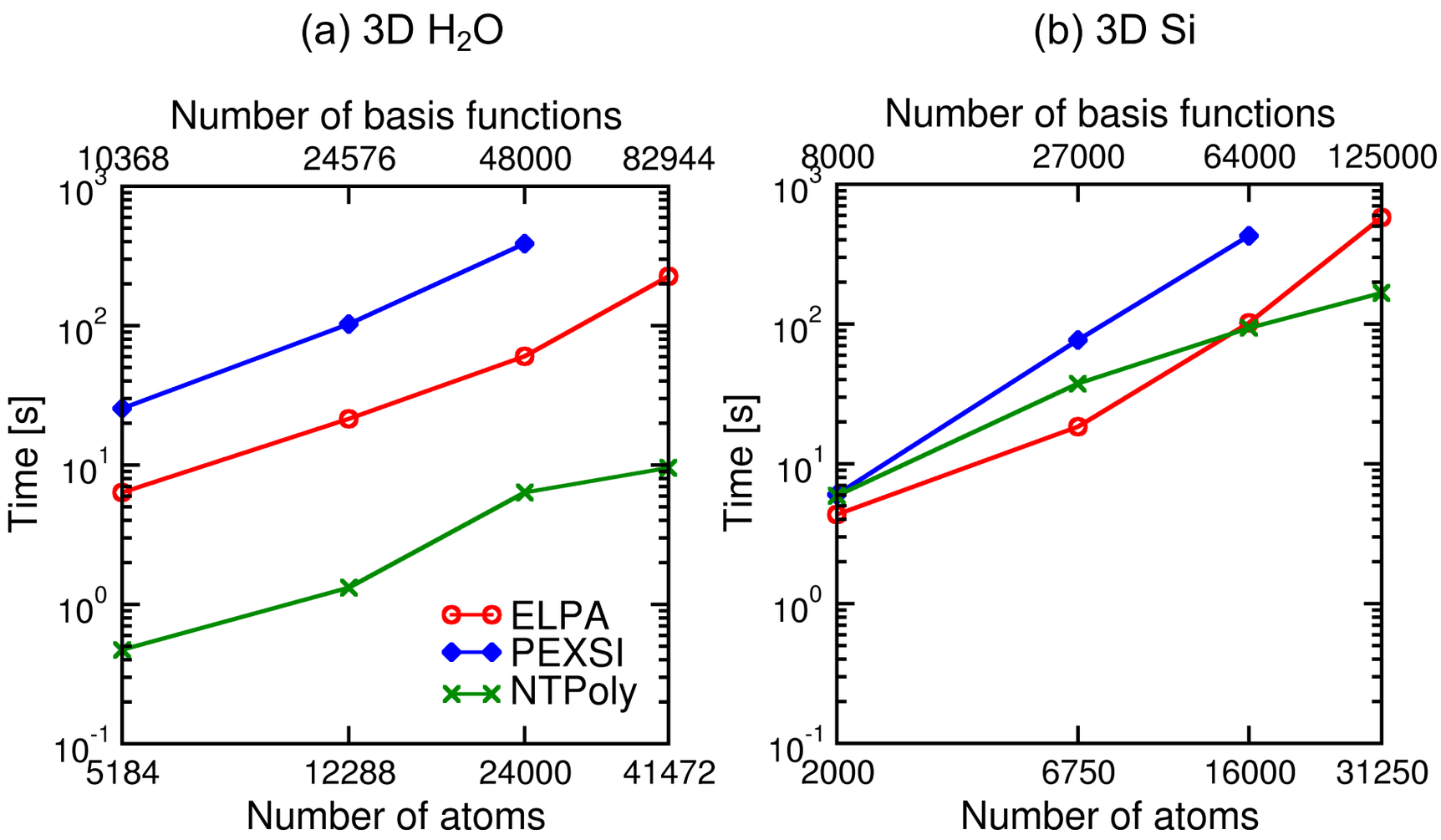
(a) H_2O



(b) Si



Example: DFTB+ (semiempirical) - ELPA, PEXSI, NTPoly



NTPoly settings:
- 4^{th} order TRS method
- 10^{-5} truncation threshold
- 10^{-2} convergence criterion

NTPoly accuracy: Band structure energies agree with ELPA within 10^{-5} eV/atom

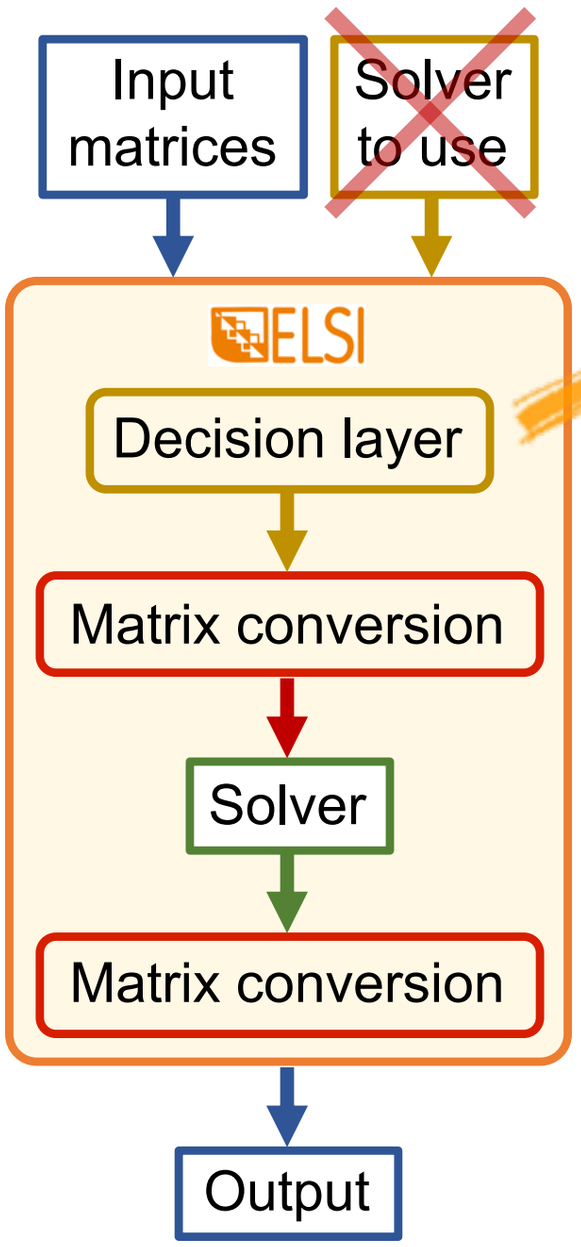
- DFTB
- DFTB+ (highly sparse matrices)
- 2,560 CPU cores on Cori-Haswell

NTPoly faster for large (sparse) gapped systems

NTPoly: Sparse Matrix Algebra, *O(N)* Solvers

Dawson, Nakajima, Computer Physics Communications 225,154-165 (2018)

ELSI Decision Layer (Beginnings)

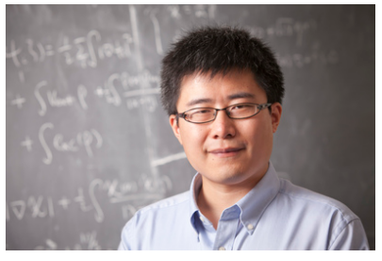


- 1st layer: Quick decision
  - Choose ELPA
    - ✓ n_basis < 20,000
  - Choose PEXSI
    - ✓ dimensionality < 3
    - ✓ sparsity > 0.95
  - Choose NTPoly
    - ✓ n_basis > 100,000
    - ✓ sparsity > 0.999
    - ✓ have_gap

- 2nd layer: Direct comparison
  - Exclude PEXSI
    - ✓ dimensionality == 3
    - ✓ n_cpu < 20,000
  - Exclude NTPoly
    - ✓ n_basis < 50,000
    - ✓ sparsity < 0.99
    - ✓ NOT have_gap
  - Try available solvers (one in an SCF step)
  - Choose fastest solver

What About Iterative Solvers? (Plane Waves)

Yingzhou Li



Jianfeng Lu (Duke)

$$\underline{\underline{h}}\,\underline{c}_k = \epsilon_k\,\underline{\underline{s}}\,\underline{c}_k$$

If $N_{basis} >> N_{ev}$, "full matrix" solvers are not competitive (time and memory).

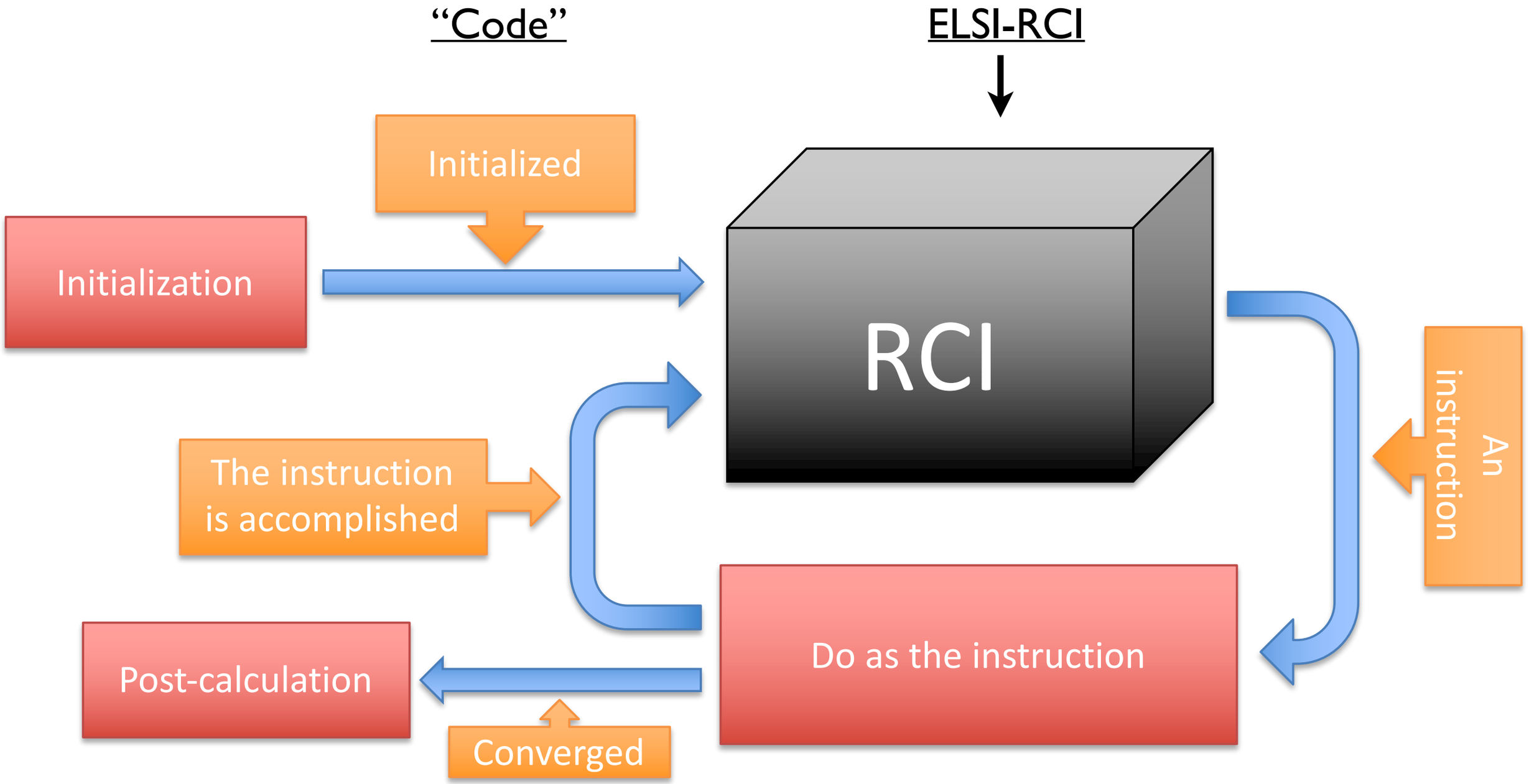
Alternative: Iterative - e.g,. Davidson

- Ψ
- Solve Rayleigh Ritz problem $(\Psi^* \mathrm{H} \Psi, \Psi^* \Psi)$ for smallest eigenpairs Λ, Q
- $R = \mathrm{H}\Psi Q - \Psi \mathrm{Q} \Lambda$
- If $\|R\| < tol$, converged
- Approximately solve $(H - \Lambda_i I)V_i = R_i$ for all V_i
- $\Psi = [\Psi\, V]$

RCI - Generic Iterative Solvers for Existing Codes

Problem: Data Structures, Distribution in Different Codes Can Vary Widely.

Solution: Ask Code to Perform Detailed Operations, Drive Sequence of Steps.



ELSI-RCI - Code Example

```
ijob = RCI_INIT_IJOB
do
  call rci_solve(r_h, ijob, iS, task, resvec)
  select case (task)
  case (RCI_NULL)
  case (RCI_STOP)
     exit
  case (RCI_CONVERGE)
     exit
  case (RCI_H_MULTI)
     call dgemm(iS%TrH, 'N', n, n_state, n, 1.0, H, n, Work(iS%Aidx)%Mat, lda, 0.0, Work(iS%Bidx)%Mat, ldb)
  case (RCI_S_MULTI)
     call dgemm(iS%TrS, 'N', n, n_state, n, 1.0, S, n, Work(iS%Aidx)%Mat, lda, 0.0, Work(iS%Bidx)%Mat, ldb)
  case (RCI_P_MULTI) ! No preconditioner
     Work(iS%Bidx)%Mat = Work(iS%Aidx)%Mat
  case (RCI_GEMM)
     call dgemm(iS%trA, iS%trB, iS%m, iS%n, iS%k, iS%alpha, Work(iS%Aidx)%Mat, iS%lda, Work(iS%Bidx)%Mat, iS%ldb, iS%beta, Work(iS%Cidx)%Mat, iS%ldc)
  case (RCI_AXPY)
     call daxpy(iS%m*iS%n, iS%alpha, Work(iS%Aidx)%Mat, 1, Work(iS%Bidx)%Mat, 1)
  case (RCI_COPY)
     Work(iS%Bidx)%Mat = Work(iS%Aidx)%
  case (…)
      …….
  end select
end do
```

call rci_solve(r_h, ijob, iS, task, resvec)

call rci_omm(r_h, ijob, iS, task, resvec)

call rci_davidson(r_h, ijob, iS, task, resvec)

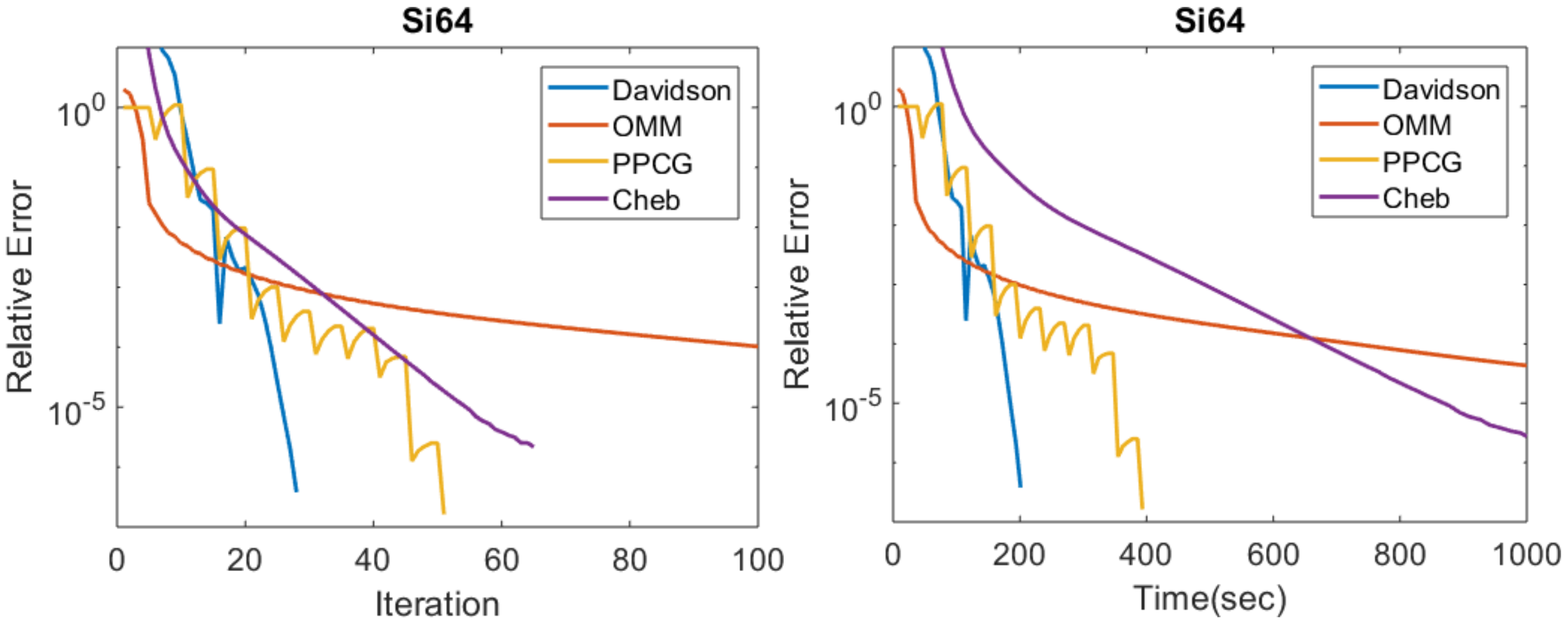
call rci_ppcg(r_h, ijob, iS, task, resvec)

Do as the instruction

ELSI-RCI - Proof of Principle

Yingzhou Li

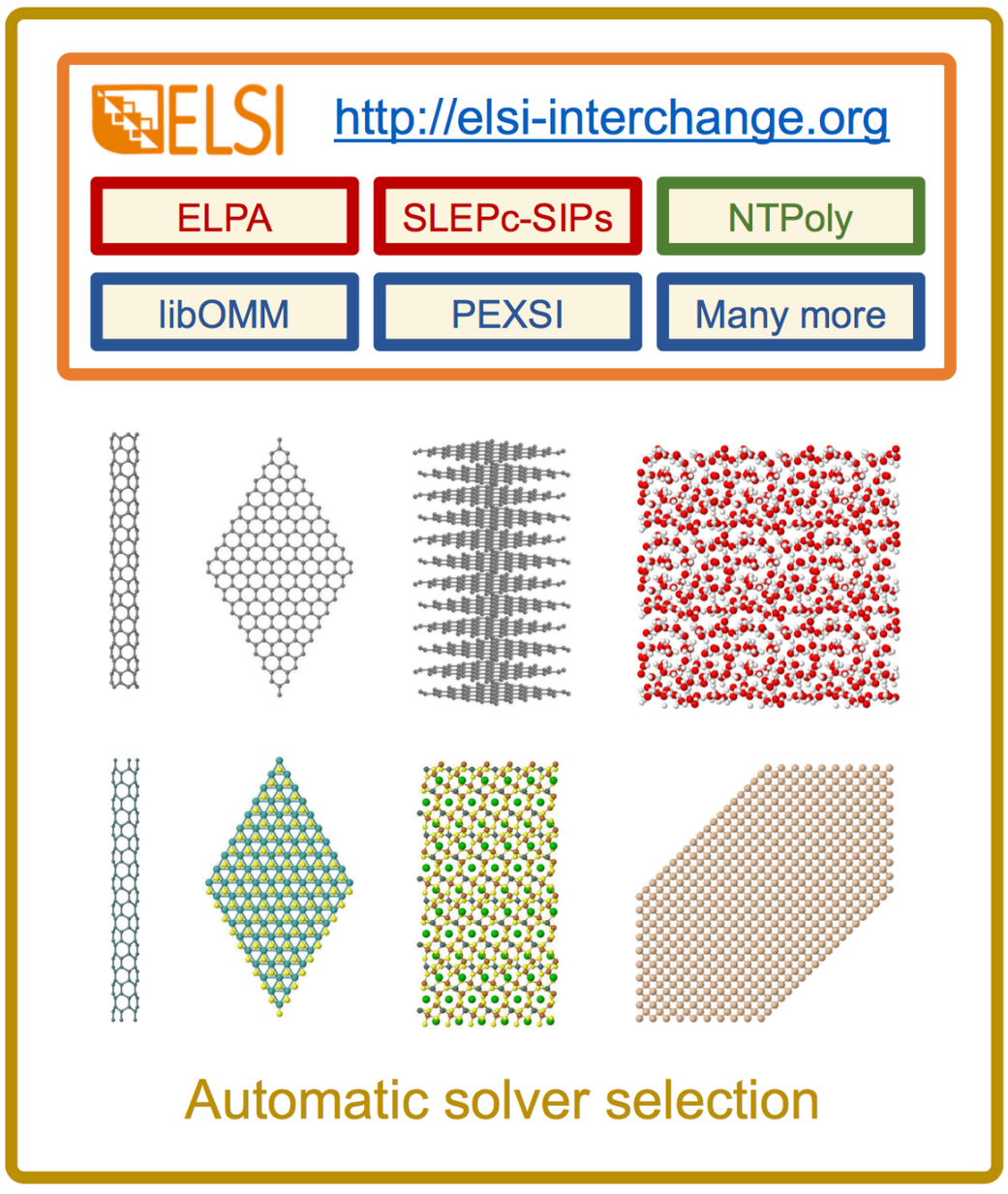
- Si supercell
- ONCV pseudopotential
- E_{cut}=20 Ha
- Kerker preconditioner
- Initial random wave function, Hamiltonian from converged SCF



→ Efficient prototyping, implementation of solvers for different purposes (e.g., BSE)

Conclusion and Acknowledgments

Victor Yu



- ELSI offers a unified interface to various eigensolvers and density matrix solvers
  - Small-to-medium calculations: ELPA
  - Large 1D/2D geometries: PEXSI
  - Large 3D geometries: NTPoly, PEXSI
  - More solvers to come

- Adopted by DFTB+, DGDFT, FHI-aims, SIESTA

- Ongoing efforts
  - RCI framework for iterative solvers
  - Solver optimization targeting GPUs

ELSI is an NSF SI²-SSI supported project under grant number 1450280. Any opinions, findings, and conclusions or recommendations expressed here are those of the author(s) and do not necessarily reflect the views of NSF.