Infrastructure Developments for Electronic Structure Codes in ELSI

NSF SI2-SSI:ACI-1450280

Volker Blum

Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC

http://aims.pratt.duke.edu

ELectronic Structure Infrastructure (ELSI) - Overview

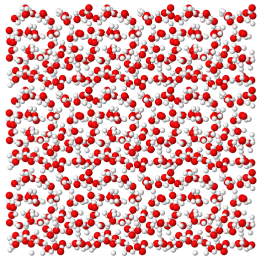
$$\underline{\underline{h}}\,\underline{c}_k = \epsilon_k\,\underline{\underline{s}}\,\underline{c}_k$$

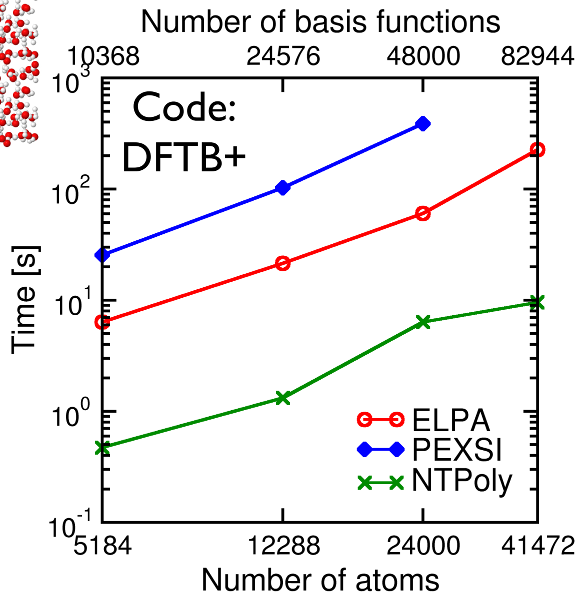
Codes:
FHI-aims, Siesta, DFTB+, DGDFT, …

ELSI

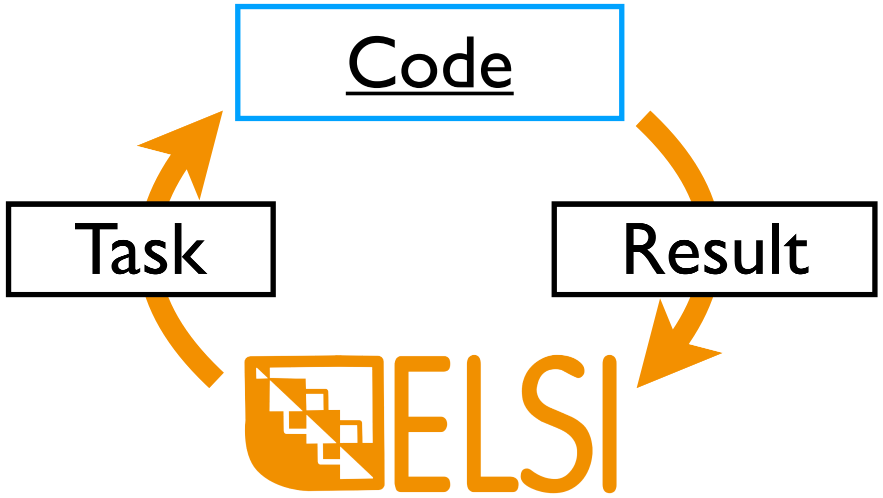
Solvers (ELPA, PEXSI, NTPoly, …)

Benchmarking Different Solvers on Equal Footing





Reverse Communication Interface for Iterative Solvers



9th International Abinit Developer Workshop - Louvain-la-Neuve, Belgium, May 22, 2019

ELSI - Acknowledgments

<u>Nucleus</u>: 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”*

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

<u>Why ELSI Works:</u>

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, …

<u>Electronic Structure Library:</u>

Electronic Structure Library cecam

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, …

My Other Day Job

http://aims.fhi-berlin.mpg.de

FHI-aims code:

- All electron
- Numeric atom-centered basis functions
- High numerical accuracy
- Non-periodic and periodic systems, equal footing
- Semilocal & hybrid DFT, *GW*, RPA, …
- Scalable to large systems at high accuracy (thousands of atoms)
- Scalable from laptops to top supercomputers
- Global community of developers and users

But when it comes to code, we still share many underlying interests.

Blum, Gehrke, Hanke, Havu, Havu, Ren, Reuter, Scheffler, Computer Physics Communications ***180****, 2175 (2009)*

The Eigenvalue Problem in Electronic Structure Theory

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

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

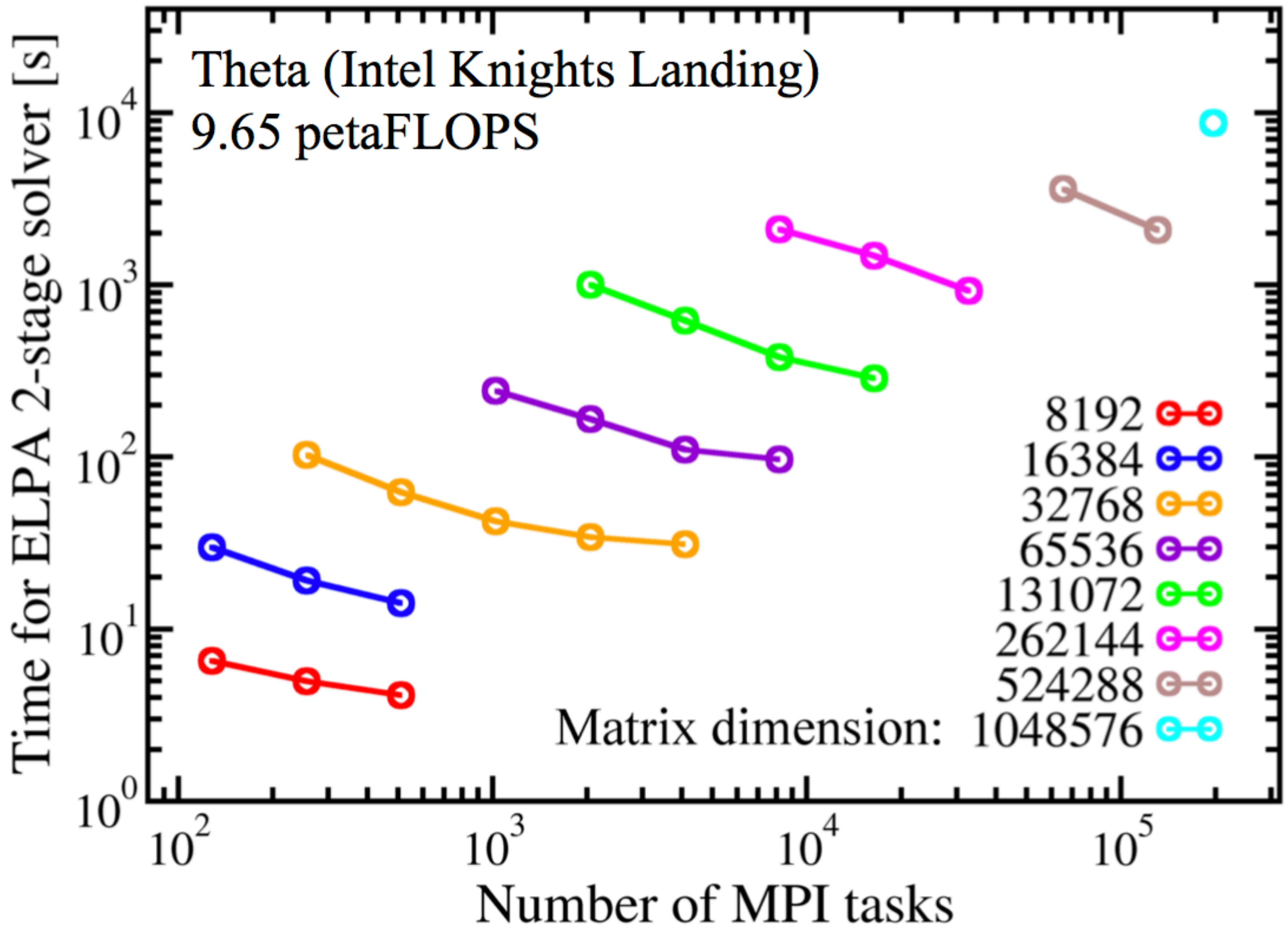
Exact solvers	Iterative solvers	O(N) solvers	Other DM-based approaches
Lapack Scalapack ELPA EigenExa Magma …	Davidson Projected Preconditioned Conjugate Gradient Chebychev Filtering Slepc-SIPS …	NTPoly Various code-internal and/or proprietary implementations	PEXSI Orbital Minimization Method FEAST …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

Scaling, 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: For Large Systems, Switch to Better Solver

Electronic structure codes

?

Solvers

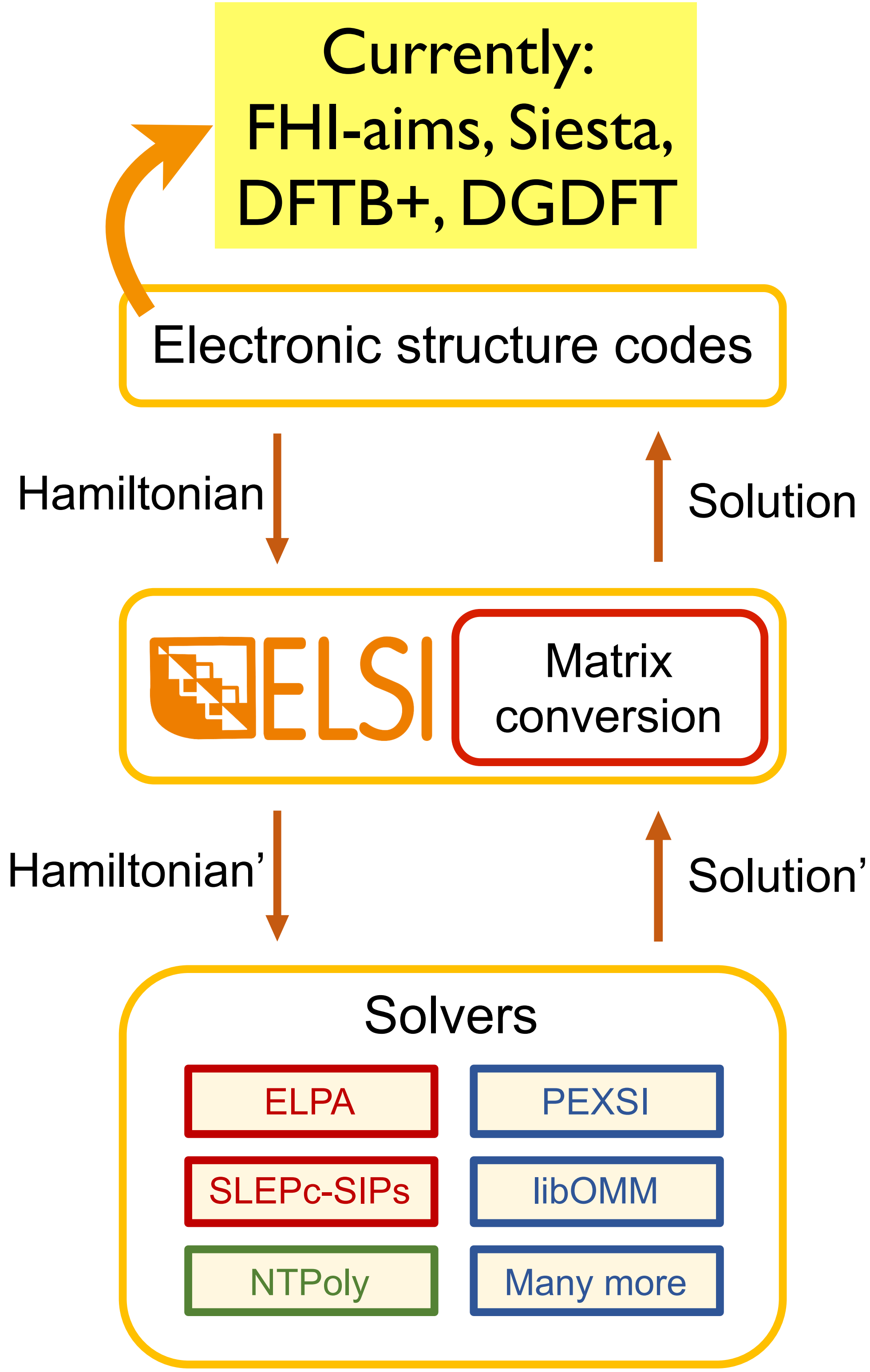
- ELPA
- PEXSI
- SLEPc-SIPs
- libOMM
- NTPoly
- Many more

? 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

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)
```

Portability, Extendability, Sustainability

- ELSI ships with an integrated build system powered by CMake
  - Two installation options: one-click vs. expert
- Supports Cray, GNU, IBM, Intel, NAG, PGI compiler suites
- Runs on platforms extending from laptops to leading supercomputers
  - Cori, Edison, K, Mira, Sierra, Summit, Titan, 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
- ELSI, Fdict, Futile, libfdf, libgridxc, libpsml, libxc, Psolver, pspio, xmlf90, …

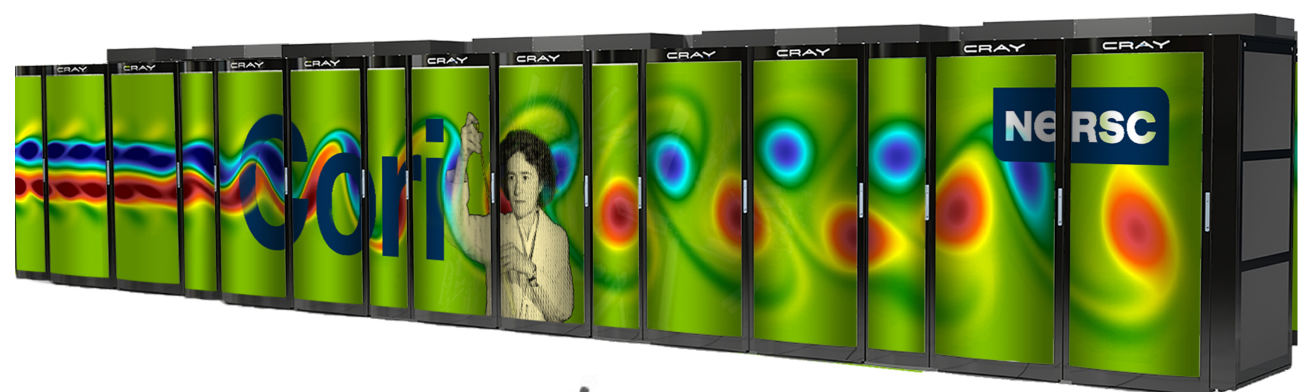
http://gitlab.com/ElectronicStructureLibrary/esl-bundle

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

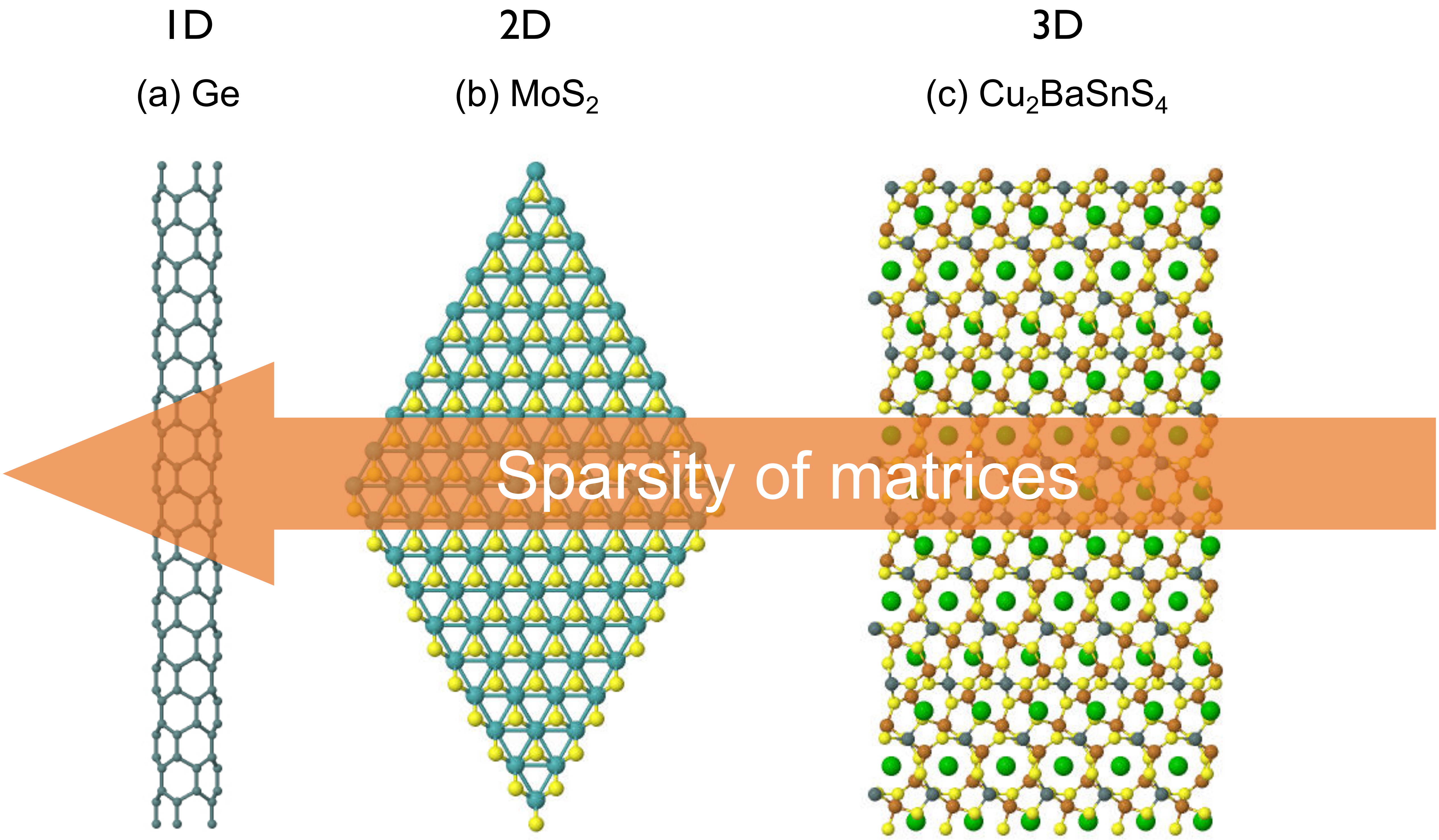
2,388 compute nodes
76,416 processing cores
2.81 Petaflops

Cori-KNL Cray XC40

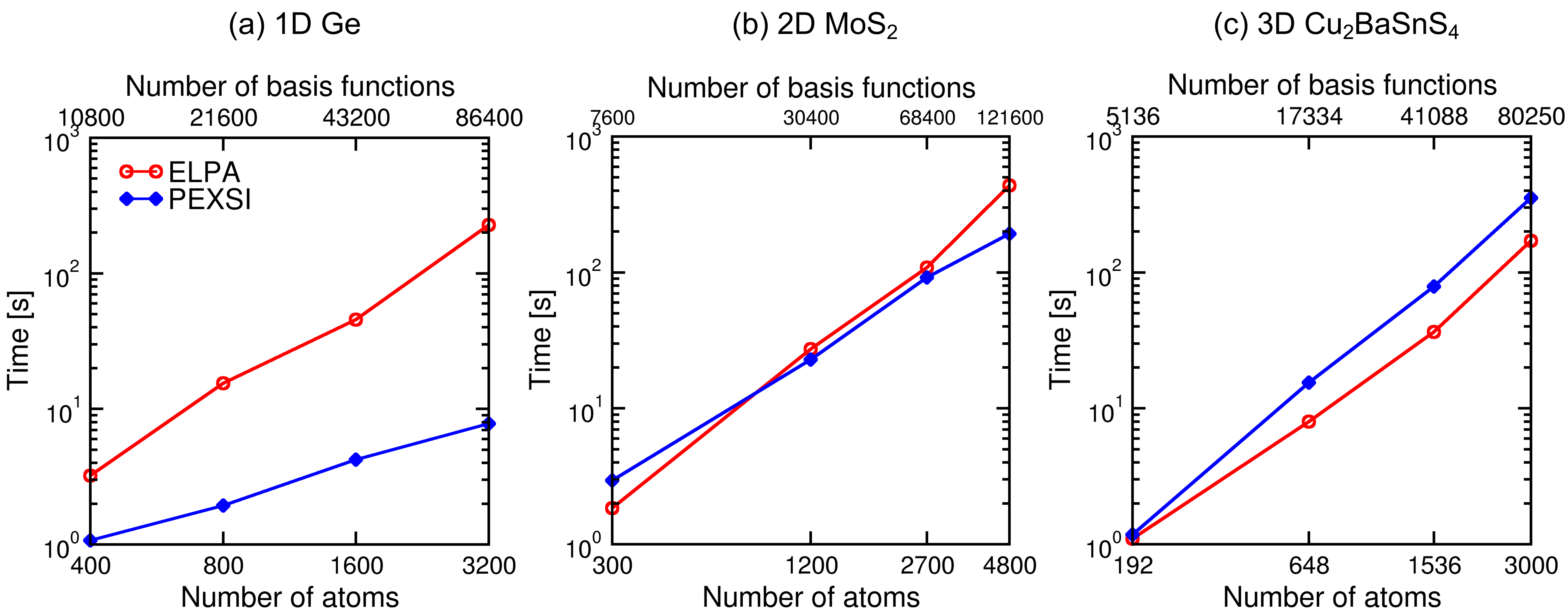
Processor: Intel Knights Landing
Interconnect: Cray Aries

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

Performance: Solver Benchmarks on Equal Footing



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



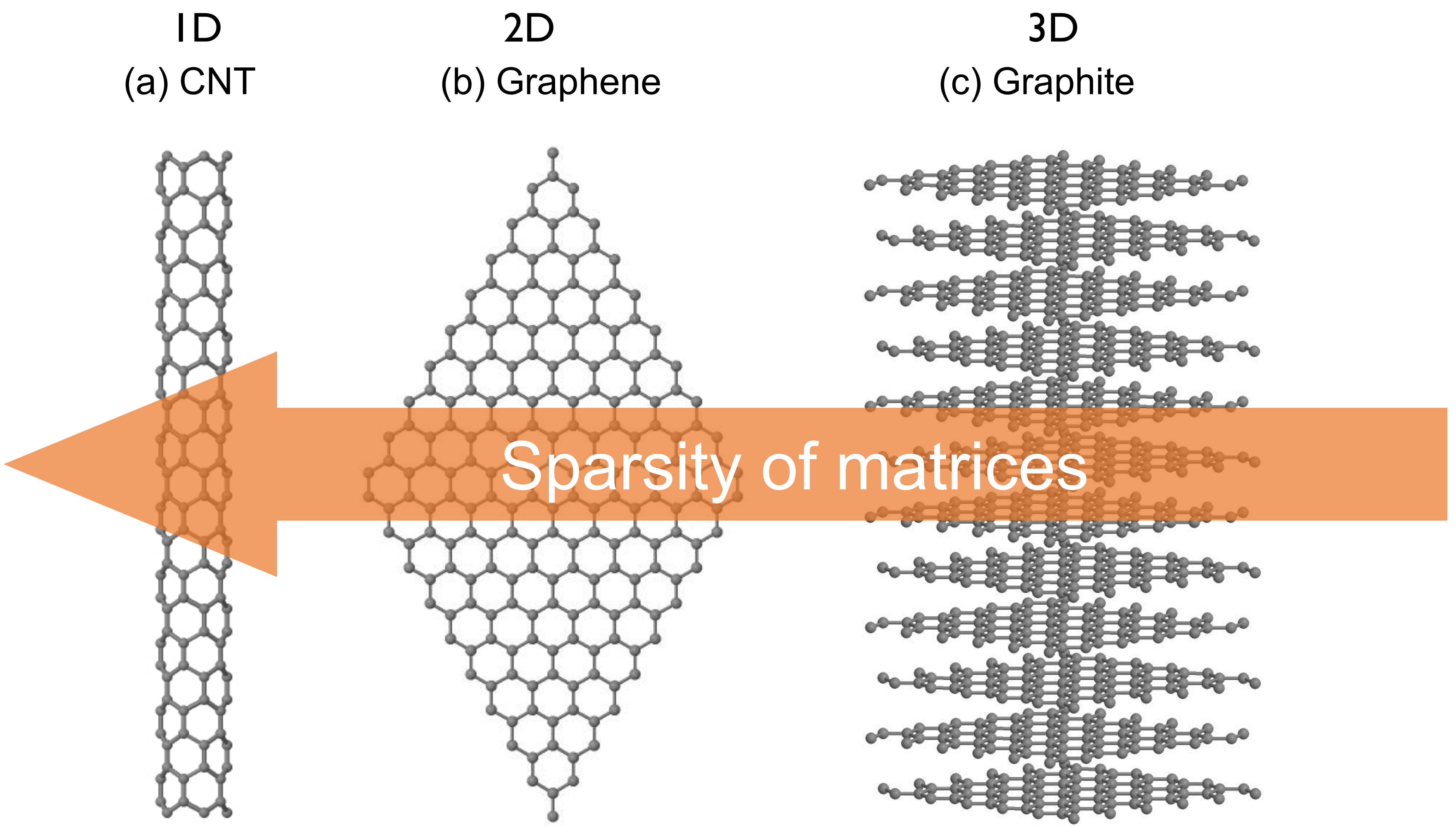
- DFT-PBE
- FHI-aims (all-electron)
- 2,560 CPU cores on Cori-Haswell

PEXSI faster for large low-dimensional (sparse) systems

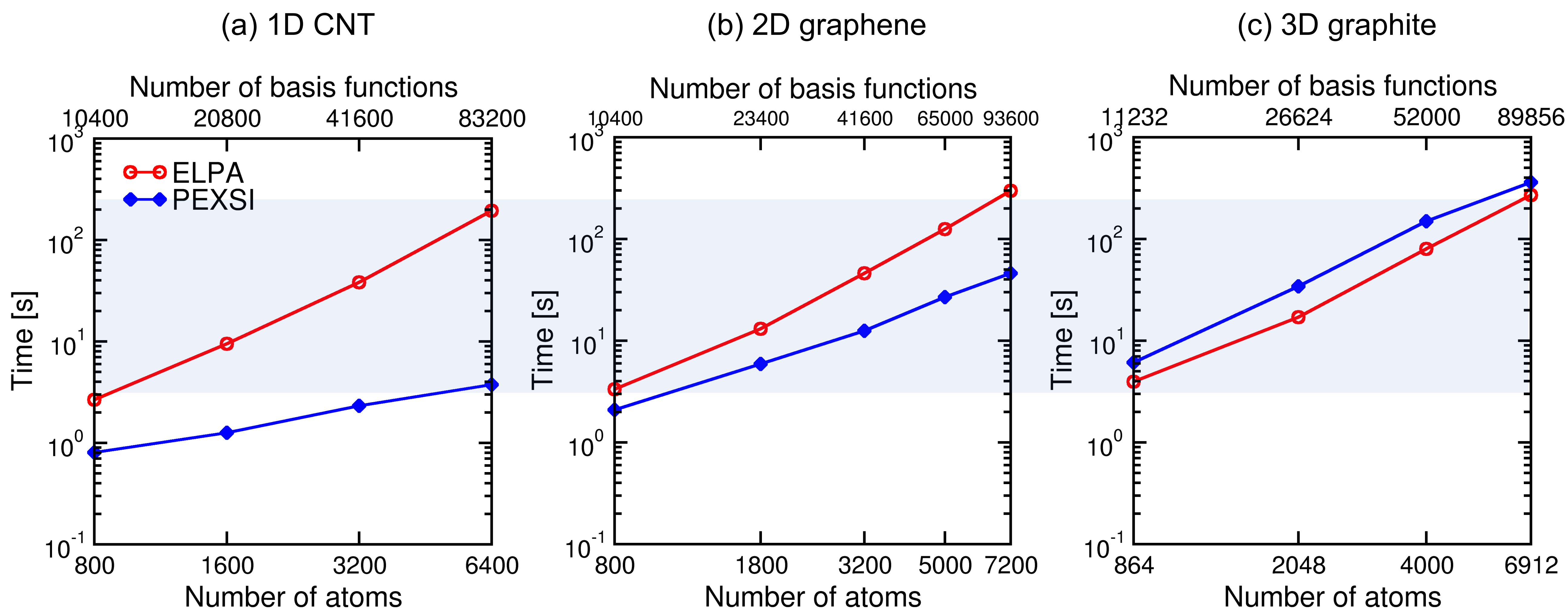
PEXSI: Semilocal DFT, $O(N)$ - $O(N^2)$ for large systems

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



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

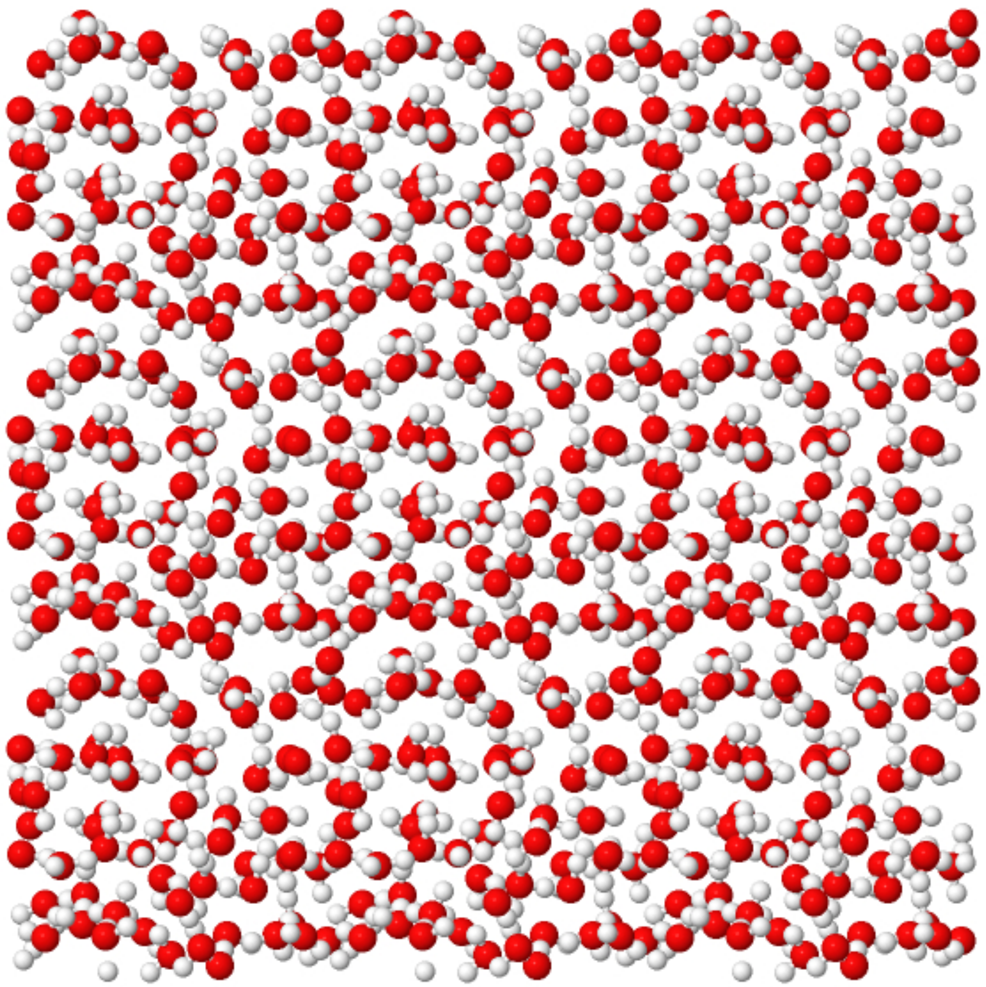
PEXSI: Semilocal DFT, $O(N)$ - $O(N^2)$ for large systems

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)

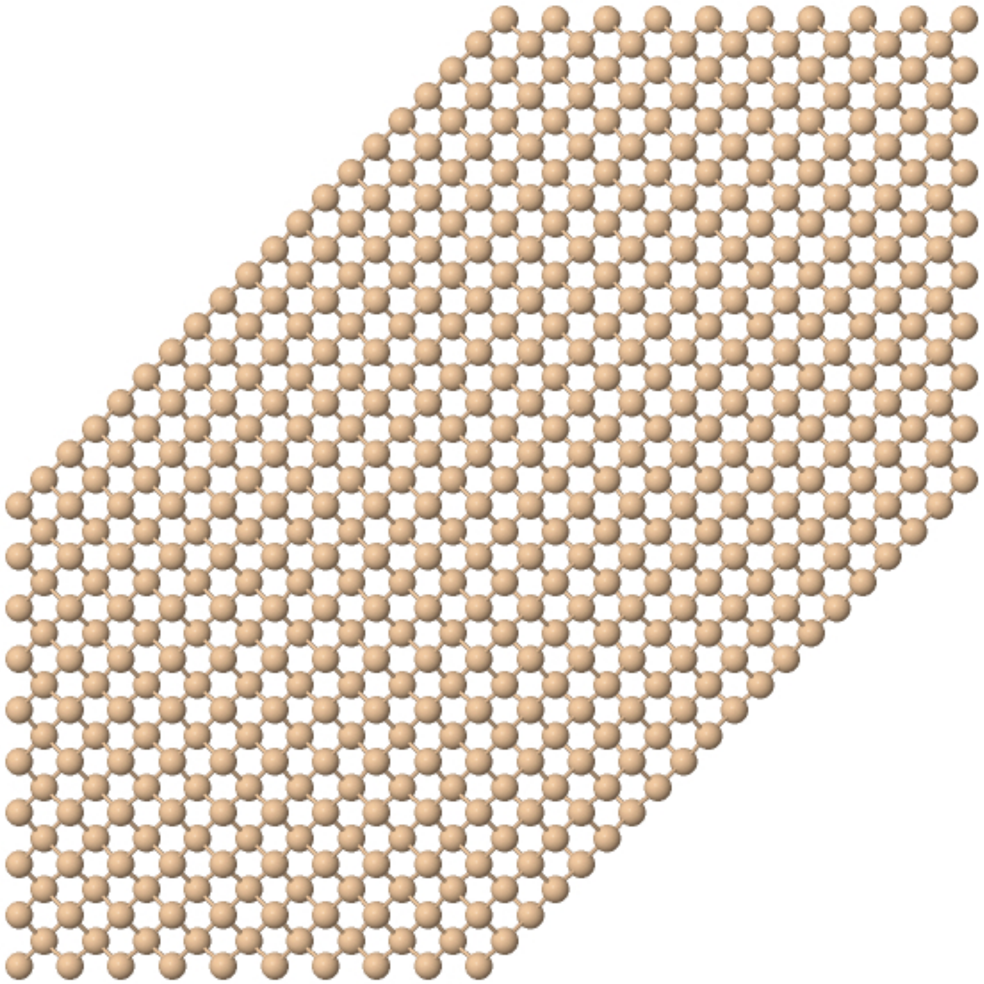
Performance: Solver Benchmarks on Equal Footing

3D Periodic Systems:

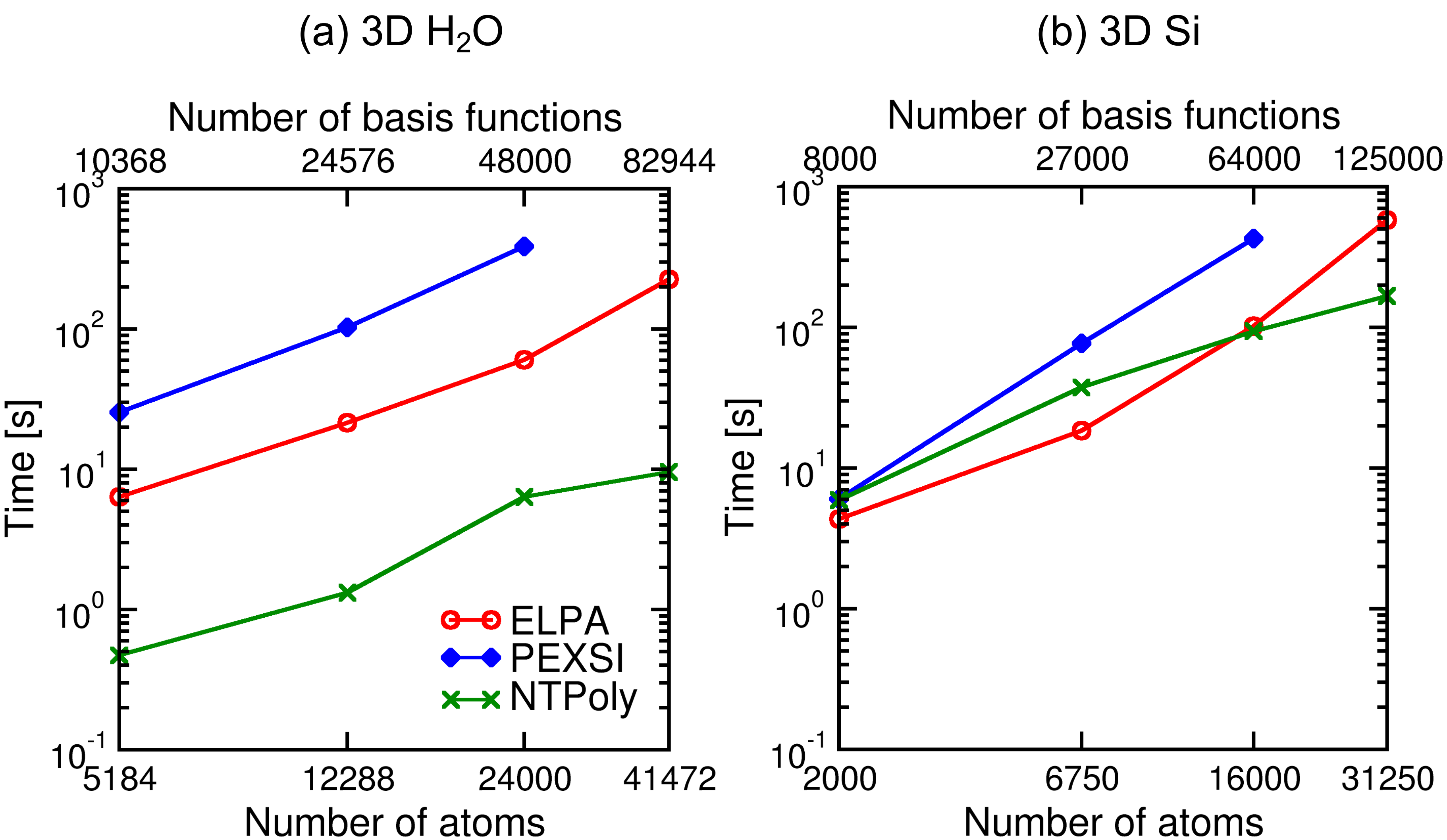
(a) H_2O



(b) Si



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



NTPoly settings:

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

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

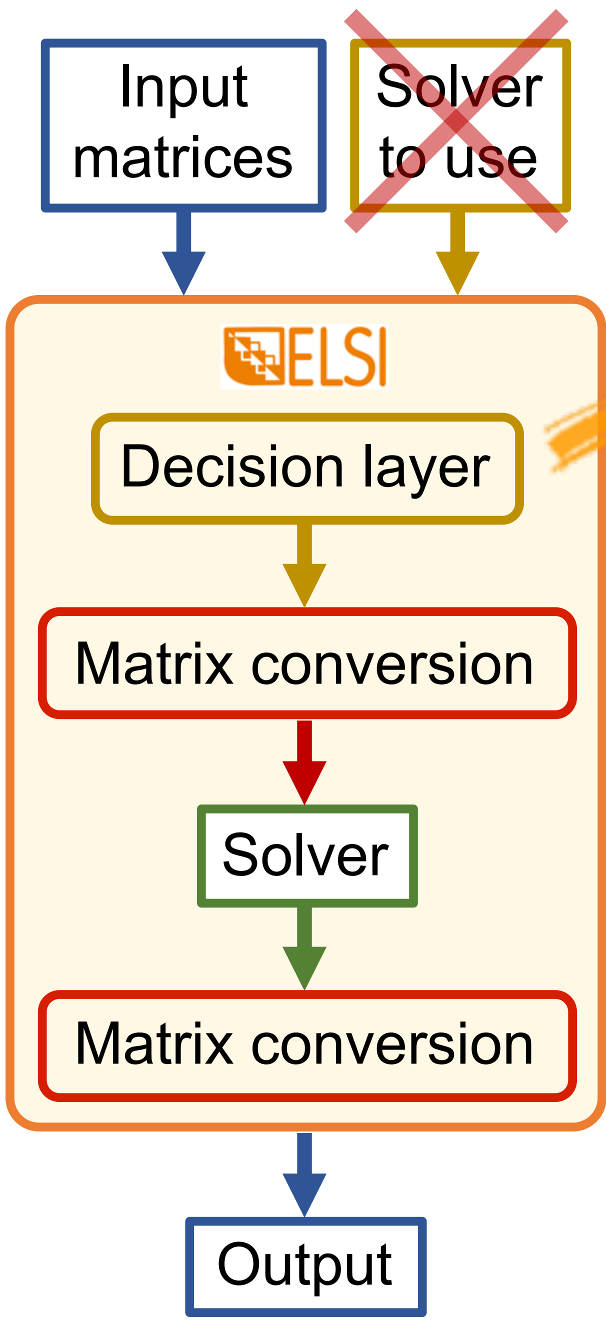
- DFTB
- DFTB+ (highly sparse matrices)
- 2,560 CPU cores on Cori-Haswell

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)

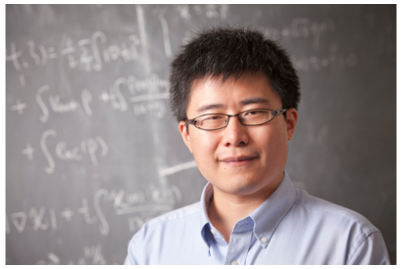


- 1st layer: Quick decision
  - Choose ELPA
    - ✓ n_basis < 20,000
  - Choose PEXSI
    - ✓ dimensionality < 3
    - ✓ sparsity > 0.95
  - Choose NTPoly
    - ✓ n_basis > 100,000
    - ✓ sparsity > 0.999
    - ✓ have_gap

- 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).

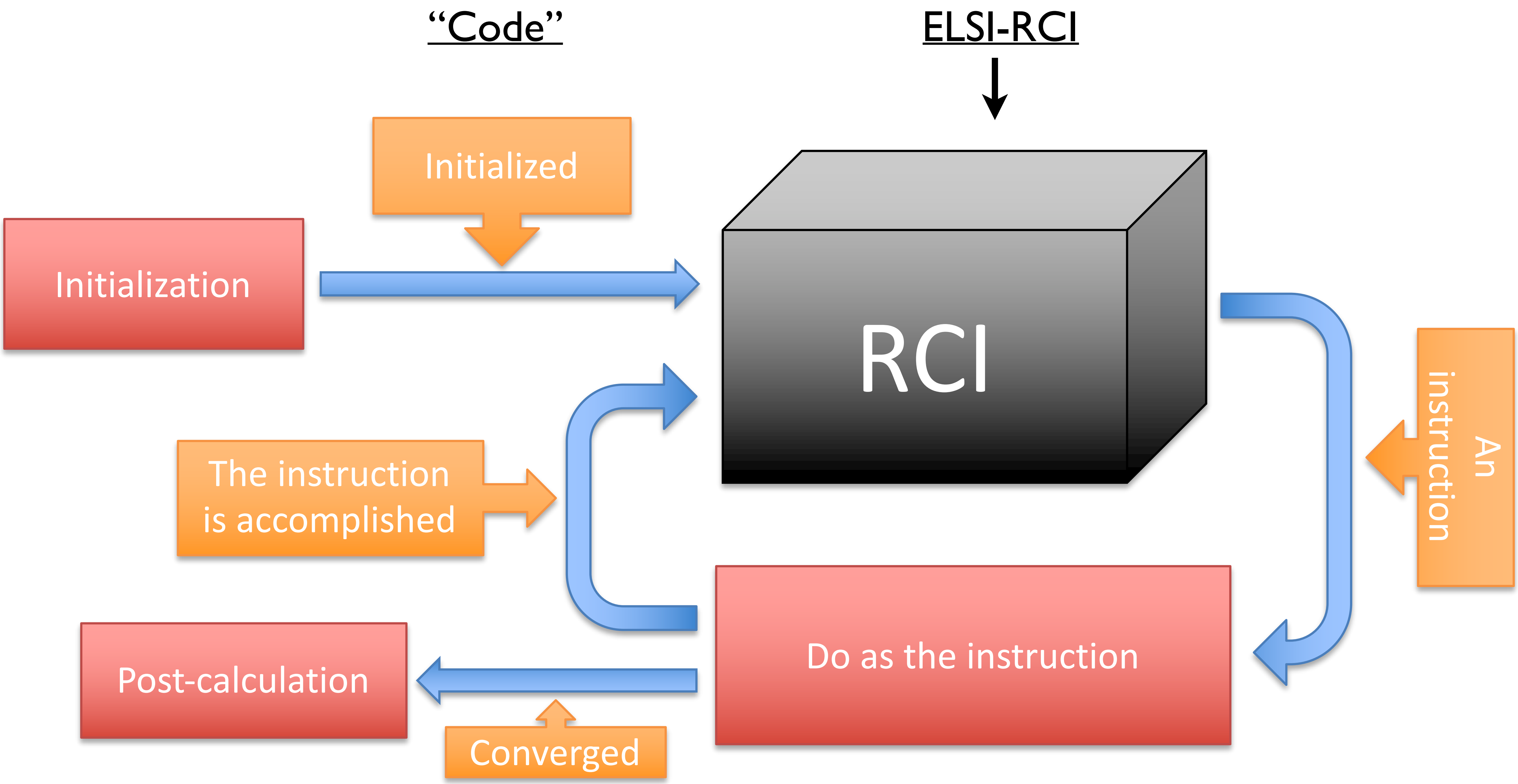
Alternative: Iterative - e.g,. Davidson

- 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_omm(r_h, ijob, iS, task, resvec)

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

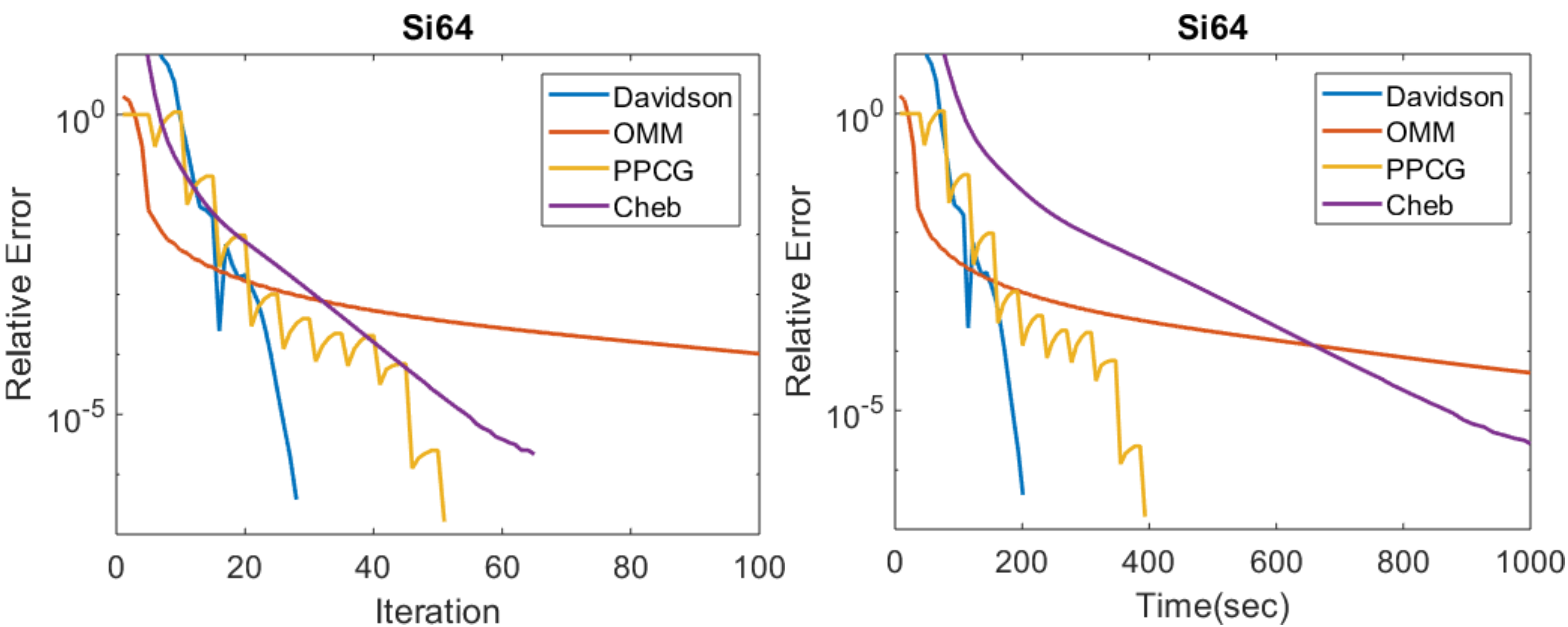
call rci_ppcg(r_h, ijob, iS, task, resvec)

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

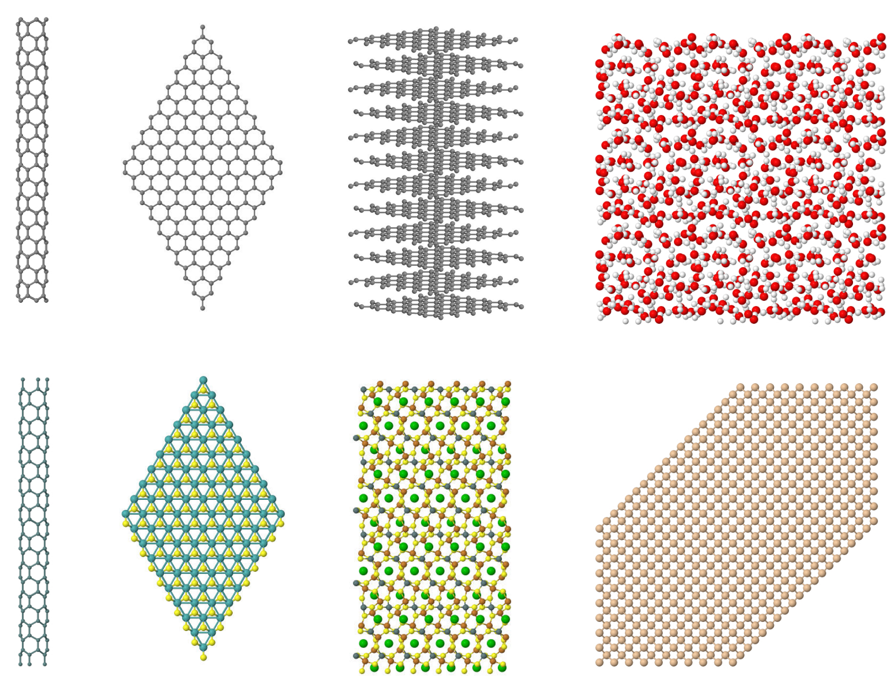


→ Efficient prototyping, implementation of solvers for different purposes (e.g., BSE)

Conclusion and Acknowledgments

ELSI http://elsi-interchange.org

ELPA	SLEPc-SIPs	NTPoly
libOMM	PEXSI	Many more



Automatic solver selection

- ELSI offers a unified software interface to a variety of high-performance eigensolvers and density matrix solvers
- Adopted by DFTB+, DGDFT, FHI-aims, SIESTA
- ELSI is intended to be an open forum, fostering international, interdisciplinary collaborations that benefit the entire community

→ Please join us at http://elsi-interchange.org

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.