Recent Developments in the ELSI Infrastructure for Large-Scale Electronic Structure Theory

Victor Yu¹, William Dawson², Alberto García³, Ville Havu⁴, Ben Hourahine⁵, William Huhn¹, Mathias Jacquelin⁶, Weile Jia⁶,⁷, Murat Keçeli⁸, Raul Laasner¹, Yingzhou Li⁹, Lin Lin⁶,⁷, Jianfeng Lu⁹, Jose Roman¹⁰, Álvaro Vázquez-Mayagoita⁸, Chao Yang⁶, and Volker Blum¹

¹Department of Mechanical Engineering and Materials Science, Duke University, USA
²RIKEN Center for Computational Science, Japan
³Institut de Ciència de Materials de Barcelona, Spain
⁴Department of Applied Physics, Aalto University, Finland
⁵Department of Physics, University of Strathclyde, Scotland
⁶Computational Research Division, Lawrence Berkeley National Laboratory, USA
⁷Department of Mathematics, University of California at Berkeley, USA
⁸Argonne Leadership Computing Facility, Argonne National Laboratory, USA
⁹Department of Mathematics, Duke University, USA
¹⁰Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València, Spain
Kohn-Sham Density-Functional Theory (KS-DFT)

Kohn-Sham equation

\[ \hat{h}_{\text{KS}} \psi_1 = \epsilon_1 \psi_1 \]

\( \hat{h}_{\text{KS}} \): Kohn-Sham Hamiltonian
\( \psi_1 \): Kohn-Sham orbitals
\( \epsilon_1 \): Eigen-energies

Basis set expansion

\[ \psi_1(\mathbf{r}) = \sum_j c_{lj} \varphi_j(\mathbf{r}) \]

\( \psi_1 \): Kohn-Sham orbitals
\( \varphi_j(\mathbf{r}) \): “Basis” functions
\( c_{lj} \): Expansion coefficients

Eigenvalue problem

\[ H \mathbf{c} = \mathbf{e} \mathbf{S} \mathbf{c} \]

\( H \): Hamiltonian matrix
\( S \): Overlap matrix
\( \mathbf{c} \): Eigenvectors
\( \mathbf{e} \): Eigenvalues

Target: Total energy is a functional of electron density: \( E = E[n] \)

- Hamiltonian \( H \) depends on electron density \( n \)
- Electron density \( n \) depends on wavefunctions \( \{\psi_1\} \)
- Wavefunctions \( \{\psi_1\} \) depend on Hamiltonian \( H \)

Solution: Repeatedly update electron density \( n \) towards self-consistency

Nonlinear optimization
“Cubic Wall” in KS-DFT

- Diagonalization with a dense eigensolver: $O(N^3)$
- Other steps: $O(N)$ (achievable with localized basis)

$N$: Number of atoms

---

**FHI-aims, DFT-PBE, graphene (2D)**

- 4,050 - 7,200 atoms
- 56,700 - 100,800 basis functions

**Time for one SCF iteration [s]**

- **Eigenvalue problem**: $O(N^3)$
- **All other steps**: $O(N)$
- **Total**

**Edison (Intel Ivy Bridge)**

- 1,920 MPI tasks

---

**Graphene (2D):**

- 4,050 - 7,200 atoms
- 56,700 - 100,800 basis functions
Conventional Solution: Diagonalization

- ELPA eigenvalue solver library: 2-stage diagonalization
- Highly efficient for solving moderately-sized eigenproblems on HPC systems

Auckenthaler et al., Parallel Comput. 2011
http://elpa.mpcdf.mpg.de
Eigensolvers and Density Matrix Solvers

Diagonalization (wavefunctions)
- Efficient for small-to-medium-sized problems
- Applicable to metals, semiconductors, insulators
- $O(N^3)$ time and $O(N^2)$ memory

... and many methods in between ...
(crossover unclear)

Traditional $O(N)$ methods (density matrix)
- Lower scaling factor and memory consumption
- Larger prefactor, overhead for small problems
- Difficulty in treating metallic systems
ELSI: Connection between KS-DFT Codes and Solvers

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

**All-electron KS-DFT**
Numeric atom-centered orbitals
Sparse matrices

**Pseudopotential KS-DFT**
Numeric atom-centered orbitals
More sparse matrices
Soler et al., J. Phys.: Condens. Matter 2002

**Semi-empirical tight-binding**
Atomic orbitals
Highly sparse matrices

**Edison supercomputer**
http://www.nersc.gov/edison

**Cray XC30**
Intel Ivy Bridge

2.57 Petaflops
5,586 compute nodes
134,064 processing cores
(24 cores per node)
Benchmark Methods

Carbon nanotube (1D)

Graphene (2D)

Graphite (3D)

Sparsity of matrices
Benchmark Methods

• All benchmarks were performed on the Edison supercomputer at the National Energy Research Scientific Computing Center (Berkeley, CA, USA).

• Compute nodes (24 CPU cores) were fully exploited by launching 24 MPI tasks per node. No OpenMP was employed.

• Third-party libraries and tools used in these benchmarks include
  • Compilers: Intel 18.0.1
  • MPI: Cray MPICH 7.6.2
  • BLAS/LAPACK/ScaLAPACK: Intel MKL 2018.1
  • ELPA: 2016.11.001
  • NTPoly: 1.3
  • PEXSI: 1.0.0
  • SLEPc: 3.8.3

• Benchmarks reported here are not official benchmarks of the above third-party libraries. Performance may differ when running different versions of code on different computers.
Pole Expansion and Selected Inversion (PEXSI)

\[ P = \sum_1 \text{Im} \left( \frac{w_1}{H - (z_1 + \mu)S} \right) \]

- **Selected inversion**: Evaluate selected elements of \((H - (z_1 + \mu)S)^{-1}\)
- **Computational cost (semilocal XC)**:
  - 1D system: \(O(N)\)
  - 2D system: \(O(N^{1.5})\)
  - 3D system: \(O(N^2)\)
- **No dependence on band gap**
- **Highly scalable**: Poles can be evaluated independently \textit{in parallel}

**P**: Density matrix  
**H**: Hamiltonian matrix  
**S**: Overlap matrix  
**\(z_1\)**: Shift (pole)  
**\(w_1\)**: Weight  
**\(\mu\)**: Chemical potential

[http://pexsi.org](http://pexsi.org)
ELPA vs. PEXSI: 1-Dimensional FHI-aims Models

- Sparsity: 96% - 99% zeros
- Theoretical scaling:
  - ELPA: $O(N^3)$
  - PEXSI: $O(N)$ for 1D systems
- PEXSI favorable for 1D systems

FHI-aims, DFT-PBE, carbon nanotube (1D)
800 - 6,400 atoms
11,200 - 89,600 basis functions

Time [s]

Number of atoms

1,920 MPI tasks
ELPA vs. PEXSI: 2-Dimensional FHI-aims Models

- Sparsity: 91% - 99% zeros
- Theoretical scaling:
  - ELPA: $O(N^3)$
  - PEXSI: $O(N^{1.5})$ for 2D systems
- Crossover: 800 atoms

**Graphene (2D)**

- FHI-aims, DFT-PBE
- 800 - 7,200 atoms
- 11,200 - 100,800 basis functions
- 1,920 MPI tasks
ELPA vs. PEXSI: 3-Dimensional FHI-aims Models

- Sparsity: 74% - 90% zeros
- Theoretical scaling:
  - ELPA: O(N^3)
  - PEXSI: O(N^2) for 3D systems
- ELPA favorable for 3D bulk systems
ELPA vs. PEXSI: Dimensionality of Systems

All calculations: FHI-aims, DFT-PBE, 1,920 MPI tasks

- **ELPA** not dependent on dimensionality
- **PEXSI** favorable for low-dimensional (sparse) systems
ELPA vs. PEXSI: 1-Dimensional SIESTA Models

- Sparsity: 97% - 99% zeros
- Same conclusion in FHI-aims, SIESTA, and DFTB+

SEISTA, DFT-PBE, carbon nanotube (1D)
800 - 6,400 atoms
10,400 - 83,200 basis functions

ELPA vs. PEXSI: 1-Dimensional SIESTA Models

Time [s]

10^{-1} 10^0 10^1 10^2 10^3

800 1600 3200 6400

1,920 MPI tasks
ELPA vs. PEXSI: 1-Dimensional DFTB+ Models

- Sparsity: > 99% zeros
- Same conclusion in FHI-aims, SIESTA, and DFTB+

DFTB+, carbon nanotube (1D)
3,200 - 25,600 atoms
12,800 - 102,400 basis functions

Time [s]

Number of atoms

1,920 MPI tasks
• Sparsity: 90% zeros

• **ELPA**: Scales up to ~20k tasks
• **PEXSI**: Almost ideal scaling

• **ELPA**: General 3D bulk systems
• **PEXSI**: Low-dimensional systems with a large number of processors
**Shift-and-Invert Parallel Spectral Transformation in SLEPc**

\[ Hc = \epsilon Sc \]

\[ (H - \sigma S)c = (\epsilon - \sigma)Sc \]

\[ S \frac{c}{H - \sigma S} = \frac{1}{\epsilon - \sigma} c \]

\[ \tilde{H}c = \tilde{\epsilon}c \]

**Eigenspectrum**

\[ \sigma_1 \quad \sigma_2 \quad \sigma_3 \quad \ldots \quad \sigma_n \]

**Processors**

\[
\begin{array}{cccc}
0 & 4 & 8 & 12 \\
1 & 5 & 9 & 13 \\
2 & 6 & 10 & 14 \\
3 & 7 & 11 & 15 \\
\end{array}
\]

Hernandez et al., ACM T. Math. Software 2005
Campos and Roman, Numer. Algorithms 2012

http://slepc.upv.es
ELPA vs. SLEPc-SIPs: 1-Dimensional DFTB+ Models

- Sparsity: > 99% zeros
- SLEPc-SIPs: Load balance across slices (MPI tasks) matters a lot

DFTB+, carbon nanotube (1D)
3,200 - 51,200 atoms
12,800 - 204,800 basis functions

ELPA vs. SLEPc-SIPs: 1-Dimensional DFTB+ Models

1,920 MPI tasks

Number of atoms

Time [s]

10^0
10^1
10^2
10^3
10^4

3200 6400 12800 25600 51200
• Sparsity: 96% - 99% zeros

• SLEPc-SIPs: Not competitive due to poor load balance

• Carbon 1s orbitals clustered in a tiny energy interval cannot be efficiently partitioned
Frozen core approximation: “Freeze” inactive core states; solve valence states only

SLEPc-SIPs can be ~ 8x faster due to an improved load balance
(Preliminary) Density Matrix Purification with NTPoly

\[ \tilde{H} = S^{-1/2} HS^{-1/2} \]

\[ \tilde{P}_0 = f_0(\tilde{H}) \]

\[ \tilde{P}_{n+1} = f(\tilde{P}_n) \]

\[ P = S^{-1/2}\tilde{P}S^{-1/2} \]

- \( \tilde{P}_{n+1} = f(\tilde{P}_n) \) often matrix polynomial of order m
- \( S^{-1/2} \) and density matrix purification available in NTPoly, powered by its sparse matrix-matrix multiplication kernel
- Currently supported purification algorithms:
  - Canonical purification (m = 3)
  - Trace resetting purification (m = 2, 3, 4, ...)
  - Generalized canonical purification (m = 3)

Dawson and Nakajima, Comput. Phys. Commun. 2018

http://github.com/william-dawson/NTPoly


Computation of Matrix Inverse Square Root

\[(I - X)^{-1/2} = I + \frac{1}{2}X + \frac{3}{8}X^2 + \frac{5}{16}X^3 + \ldots\]

Convergence: \[\|X\|_2 = \|\lambda S - I\|_2 \leq 1\]

Newton-Schulz method + Taylor expansion

Higham, Numer. Algorithms 1997
Niklasson, Phys. Rev. B 2004

DFTB+, silicon (3D)
6,750 - 54,000 atoms
27,000 - 216,000 basis functions

Time [s]

1,920 MPI tasks

Number of atoms
ELPA vs. NTPoly: 3-Dimensional DFTB+ Models

- Sparsity: > 99% zeros
- Theoretical scaling:
  - ELPA: $O(N^3)$
  - NTPoly: $O(N)$
- Tests with other purification algorithms ongoing
Conclusions and Outlook

ELSI offers a seamless connection between electronic structure software and a variety of solver libraries.

- **Solvers:** ELPA, LAPACK, libOMM, NTPoly, PEXSI, SLEPc-SIPs
- **Parallel matrix I/O, matrix visualization, chemical potential, ...**

Based on our benchmarks and analysis, we recommend:

- An optimized dense eigensolver (ELPA) for small-to-medium-sized calculations;
- PEXSI for large, low-dimensional geometries;
- (Work-in-progress) Density matrix purification for large, bulk, gapped systems.

Future work directions include:

- More solvers: Iterative solvers (RCI), FEAST, Chebyshev filtering, and more!
- More users: Integration of ELSI into more electronic structure code projects.
Acknowledgments

http://elsi-interchange.org

Developers and contributors:

Volker Blum (Duke Univ.)       William Huhn (Duke Univ.)       Jianfeng Lu (Duke Univ.)
Danilo Brambila (FHI)          Mathias Jacquelin (LBL)         Wenhui Mi (Duke Univ.)
Christian Carbogno (FHI)       Weile Jia (LBL)                  Stephan Mohr (BSC)
Fabiano Corsetti (QuantumWise) Murat Keçeli (ANL)          Jose Roman (UPV)
William Dawson (RIKEN)         Florian Knoop (FHI)              Ali Seifitokaldani (Duke Univ.)
Alberto García (ICMAB-CSIC)    Björn Lange (Duke Univ.)         Álvaro Vázquez-Mayagoitia (ANL)
Stefano de Gironcoli (SISSA)   Raul Laasner (Duke Univ.)        Chao Yang (LBL)
Ville Havu (Aalto Univ.)       Yingzhou Li (Duke Univ.)        Haizhao Yang (Duke Univ.)
Ben Hourahine (Strathclyde Univ.) Lin Lin (LBL)           Victor Yu (Duke Univ.)

ELSI is an NSF SI2-SSI supported software infrastructure 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.