Parallel Shift-Invert Spectrum Slicing for Symmetric Self-Consistent Eigenvalue Computation

David Williams-Young, Chao Yang
Scalable Solvers Group
Computational Research Division
Lawrence Berkeley National Lab

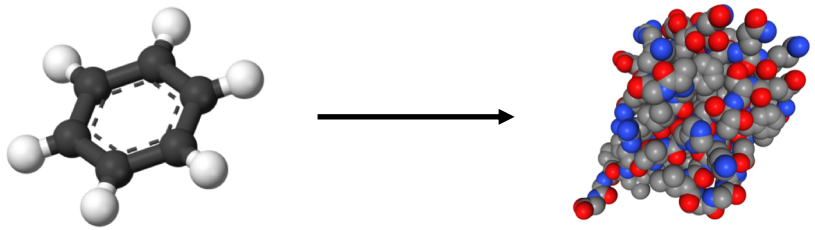
SIAM Conference on Parallel Processing for Scientific Computing (PP20)

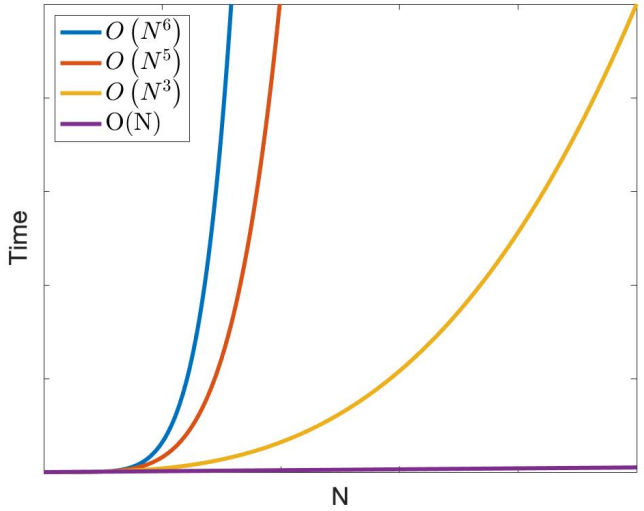
Electronic Structure Theory is the Study of Molecular Properties

Able to probe:
- Molecular structure
- Spectroscopies
- Reaction Kinetics / Mechanism

Variable cost-to-accuracy ratio:
- Density Functional Theory (DFT): $O(N^3)$
- Perturbation Theory: $O(N^5)$
- Coupled Cluster $O(N^6)$

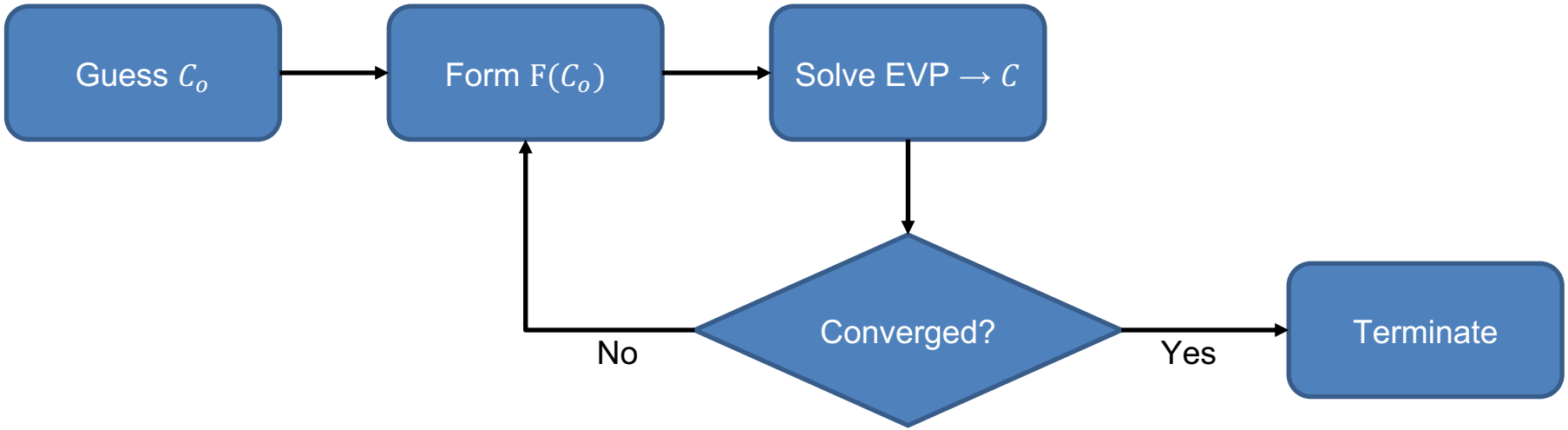
- **We want to make these simulations faster!**





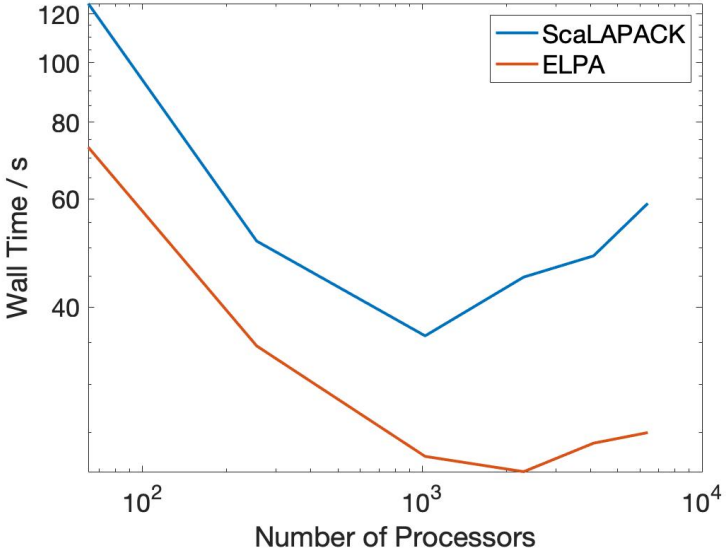
Diagonalization is the Bottleneck for Large Scale DFT Calculations

$$F(C_o)C = SCE : O(N^3) \quad \leftrightarrow \quad \min_{C_o} \mathcal{E}(C_o)$$



Direct Diagonalization Methods Exhibit Poor Scaling

PDSYEV on Cori Haswell



$N = 17{,}077 \qquad NB = 128$

$$FC = SCE: O(N^3)$$

Direct Methods:

- (Sca)LAPACK / ELPA
- $N = O(10^5)$ using $O(10^3)$ CPUs

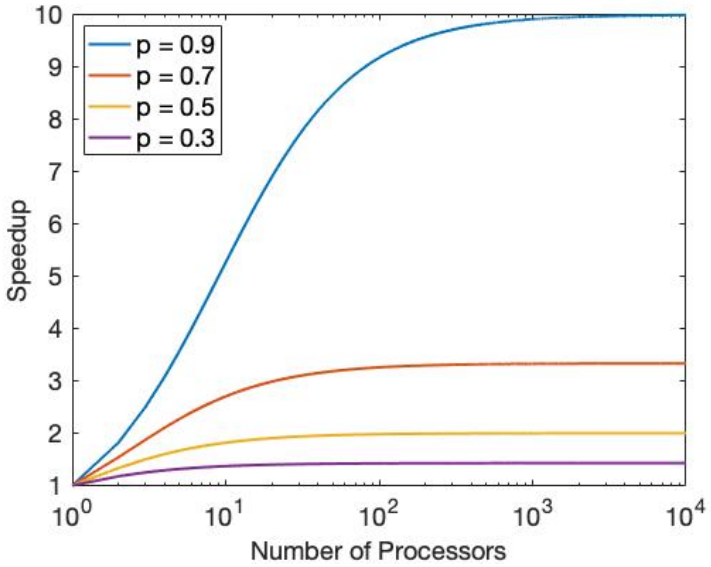
✓ "Exact"

✗ Heavily communication bound

✗ Unable to exploit sparsity

✗ Not well suited for GPUs ($N \approx 20{,}000$)

Krylov Subspace Methods are Heavily Bound By Amdahl's Law



$$FC_o = SC_oE_o : O(N^3)$$

Krylov Subspace Methods:

- Lanczos / LOBPCG / GPLHR
- $N = O(10^7)$ using $O(10^4)$ CPUs

✓ Targeted determination of C_o

✓ Only requires $Y \leftarrow FX$

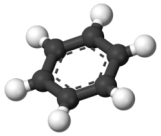
✗ Requires direct method for projected problem

✗ Many synchronization points → computational bottlenecks

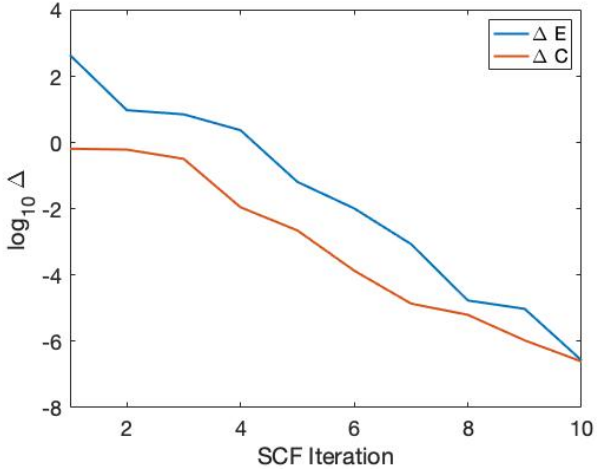
General Methods are for General Problems

We Must Exploit the Structure of the SCF Problem and Aspects of Modern Computing Architectures to Obtain Performance Improvements

Accuracy May be Amortized Over the SCF Procedure



NWChemEx SCF Convergence



- Accuracy in C_o is only needed to the same order as ΔC / ΔE

$$\left\| \left(I - C_o^{(i-1)} C_o^{(i-1),T} \right) C_o^{(i)} \right\|_2 < \left\| \left(I - C_o^{(i)} C_o^{(i),T} \right) C_o^{(i+1)} \right\|_2$$

- Smooth convergence indicates that C_o may be used to seed subsequent iterations

FLOPs are Cheap vs Communication

Reliance on accelerators (GPUs):

- IBM POWER9: 1 TFLOP/s
- Intel KNL: 3 TFLOP/s
- **NVIDIA Tesla V100: 7.8 TFLOP/s**
  - **46.8 TFLOP/s / Summit node (6x)**

Cheap FLOPs → Exposing Communication

- NVLink: peak (theoretical) 100 GB/s
- PCI-E: ~1-2 GB/s

E.g. 10k x 10k DGEMM on V100 (PCI-E)

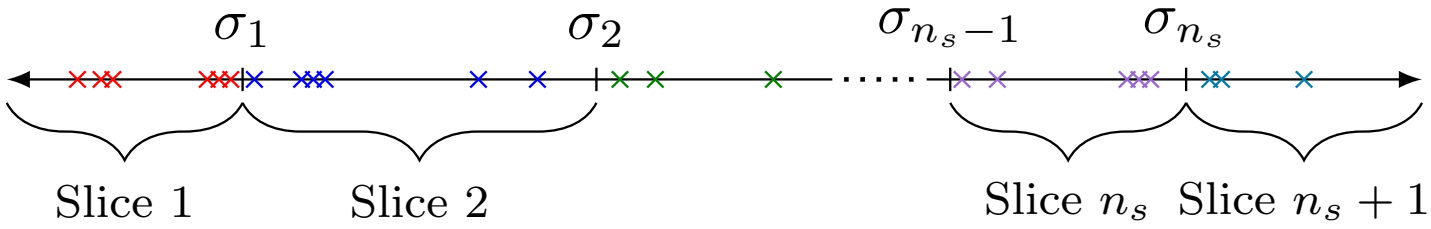
- **DGEMM time = 2 sec**
- **Communication (H2D + D2H) = 6 sec**

The SISLICE Method

A Parallel Implementation of Shift-Invert Spectrum Slicing for the SCF Eigenvalue Problem

arXiv:1908.06043 (submitted to ACM TOMS)

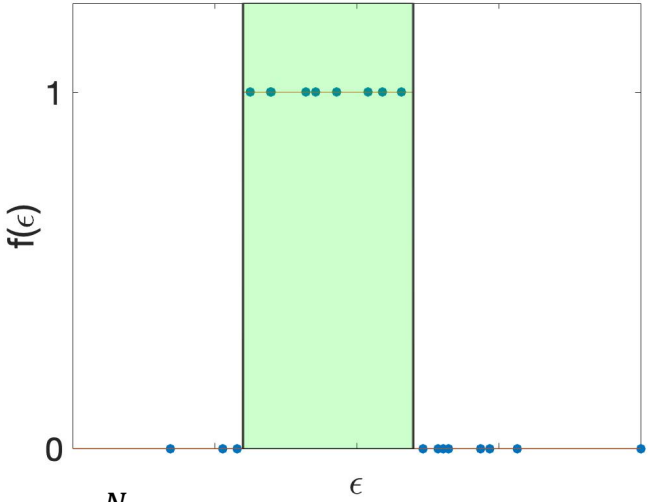
Spectrum Slicing Partitions the Eigenspectrum into Independent Tasks



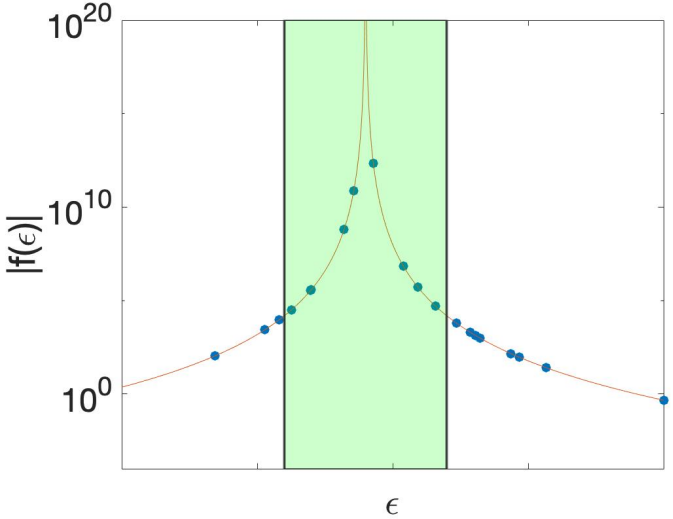
- Eigenvalues in each slice are to be determined "independently"
- Trades redundant FLOPs for less communication (ideally)

Spectral Filtering Provides a Mechanism for Spectrum Partitioning

$$f(\varepsilon) = \Theta(\varepsilon - \varepsilon_{min}) - \Theta(\varepsilon - \varepsilon_{max})$$



$$f(\varepsilon) = \frac{1}{\varepsilon - \sigma}$$



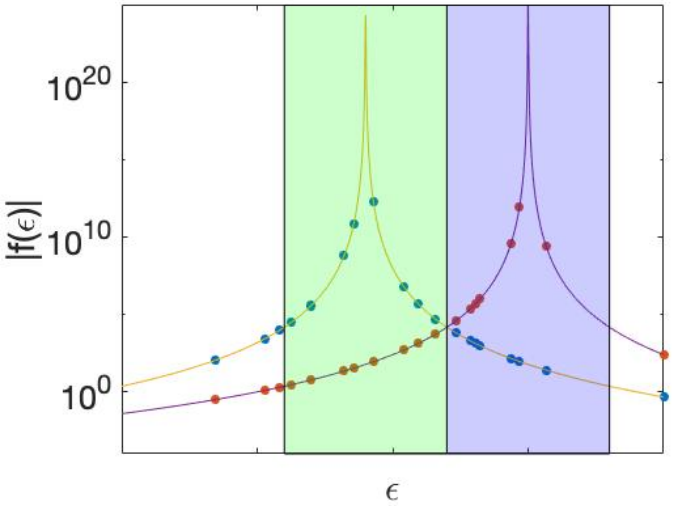
$$f(F) = \sum_{i=1}^{N} f(\varepsilon_i) C_i C_i^T$$

$$X_{k+1} = \mathrm{orth}(f(F)X_k)$$

$X_k \rightarrow C_i$'s with largest $|f(\varepsilon_i)|$

Convergence Rate for $C_i \propto |f(\varepsilon_i)|$

Shift-Invert Spectrum Slicing Trades Diagonalization for Redundant Linear System Solves



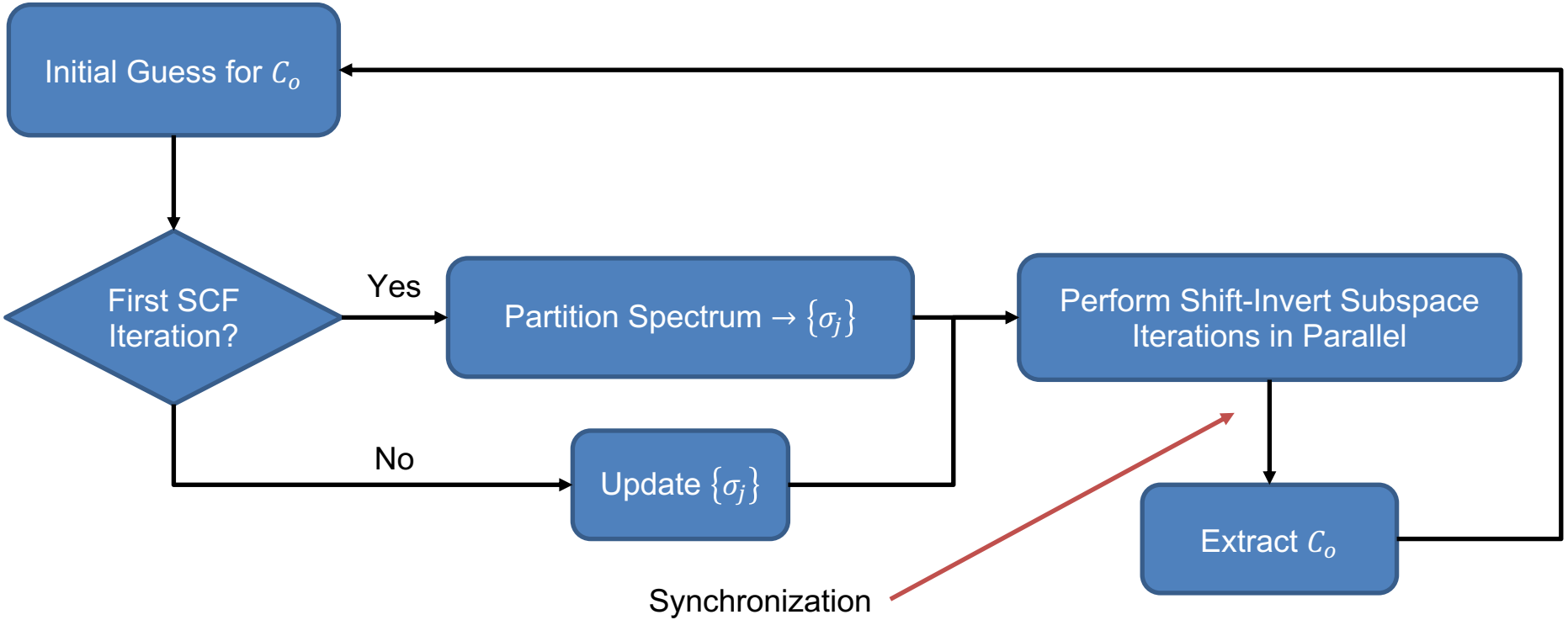
$$f(F) = (F - \sigma S)^{-1} S : O(N^3)$$

- Triangular Factorization (LU / LDLT) + Back Solve

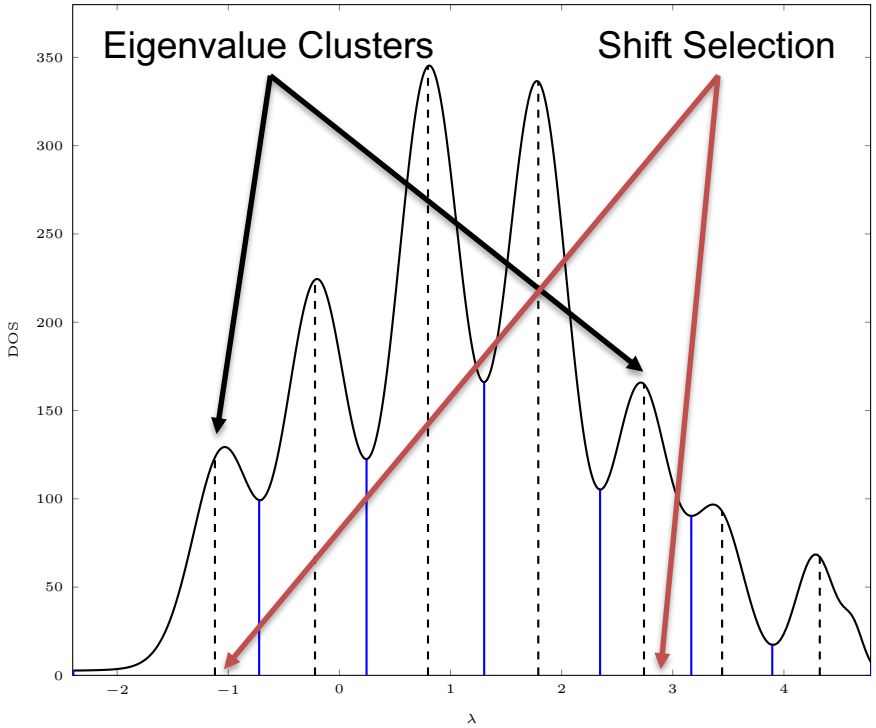
✓ Lower prefactor / better strong scaling
✓ Able to exploit sparsity (SuperLU, symPACK, etc)
✓ Suitable for GPUs
✗ Orders of magnitude more FLOPs

✗ Implementations of shift-invert spectrum slicing have struggled with expressing ***massive*** concurrency
  - Sequential shift selection

The SISLICE Method Exploits the Convergent Properties of the SCF Procedure



Density of States Estimation Allows for Better Concurrency in Spectrum Slicing



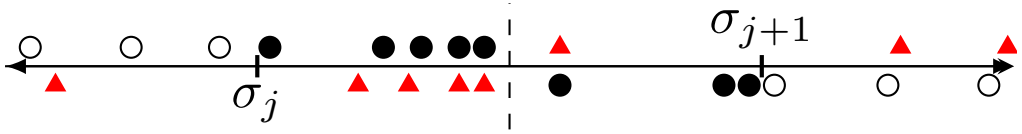
Lin, L., et al. SIAM Review, 58(1), 2016, 34–65
DBWY, et al**. arXiv:1908.06043**

- Eigenvalue distribution (DOS) may be approximated using a k-step Lanczos procedure

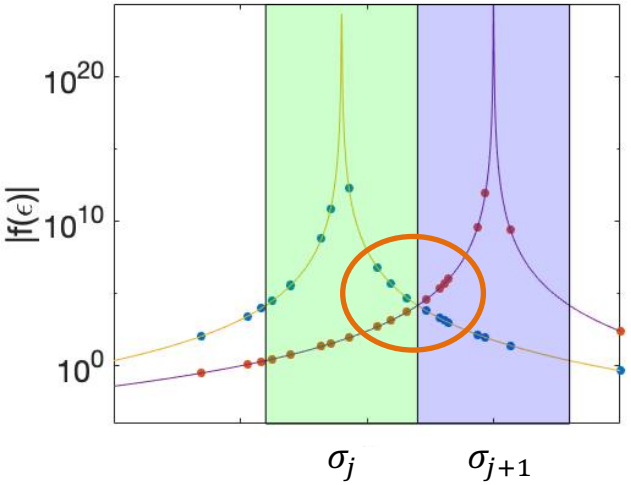
$$c(\lambda) = \sum_{l=1}^{k} d_l \exp(-\alpha_l(\lambda - \theta_l)^2)$$

- Shift are selected prior to shift-invert iterations
  - **All sets of shift-invert may be done concurrently!**

Matrix Inertia Enables Validation of Eigenpairs in Overlapping Spectral Regions



- Valid eigenpair approximations
- Rejected eigenpair approximations
- Possible eigenpair approximations based on σ_{j-1} and σ_{j+2}

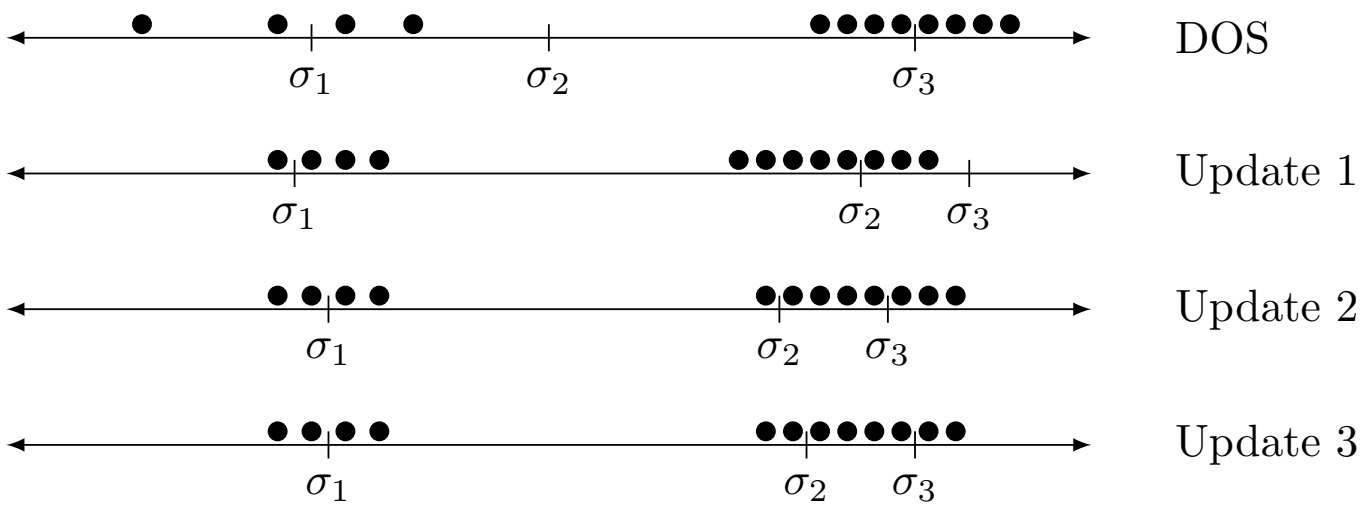


SYTRF: $F - \sigma_j S = L_j D_j L_j^T$

- Eigenvalue count between σ_j and σ_{j+1} : $n_-(D_{j+1}) - n_-(D_j)$
- Keep eigenvalue approximations with smallest residual

DBWY, et al. **arXiv:1908.06043**

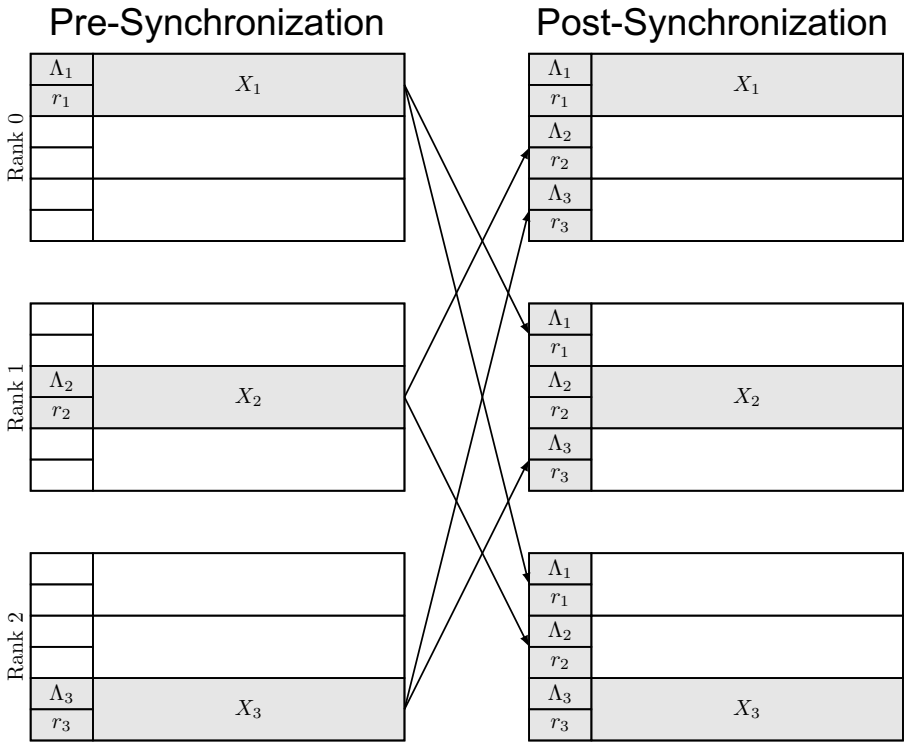
Eigenvalue Clustering Enables Systematic Improvements in Shift Placement Throughout the SCF Procedure



- Approximate eigenvalues are clustered (k-means) at each SCF iteration
- Optimal shift selection is amortized over the SCF procedure

DBWY, et al. **arXiv:1908.06043**

Slice Migration and Validation Schemes Drastically Reduce Communication Volume



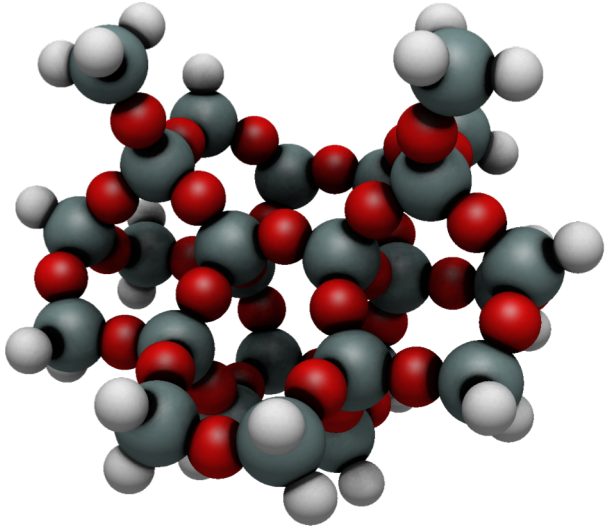
- Eigenvalue validation and shift selection methods only require scalar quantities:
  - Eigenvalue approximations
  - Residual norms
  - Matrix inertia

- **Can be replicated on each rank with no subsequent communication**
  - Synchronization latency may be hidden with RMA

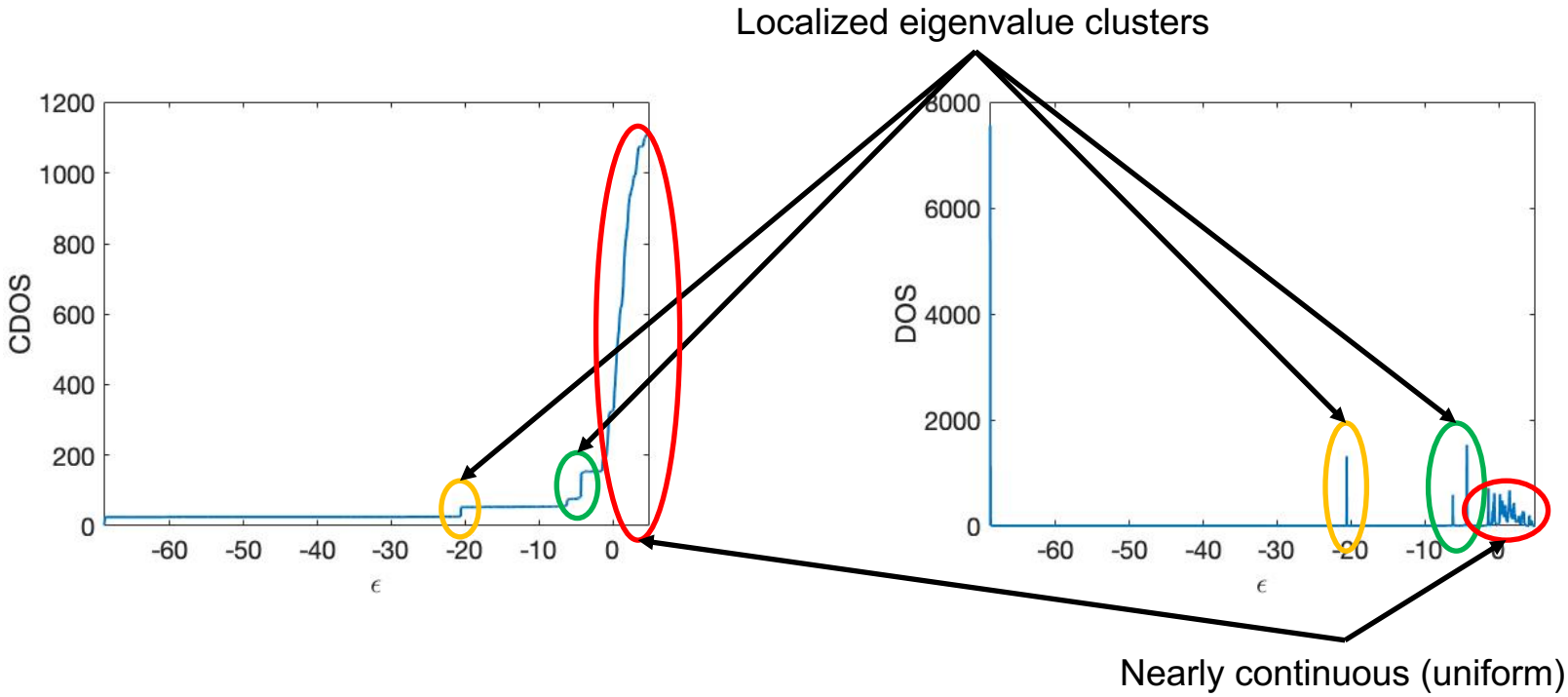
DBWY, et al. **arXiv:1908.06043**

Numerical Experiments: $SiOSi_5$

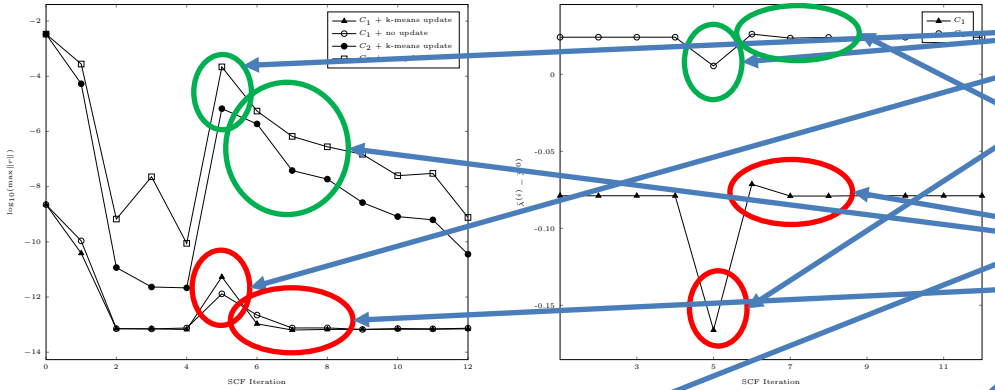
- RPBE0 / 6-31G(d) – All electron simulation
- $N = 1109$
- SCF convergence is sensitive to accuracy

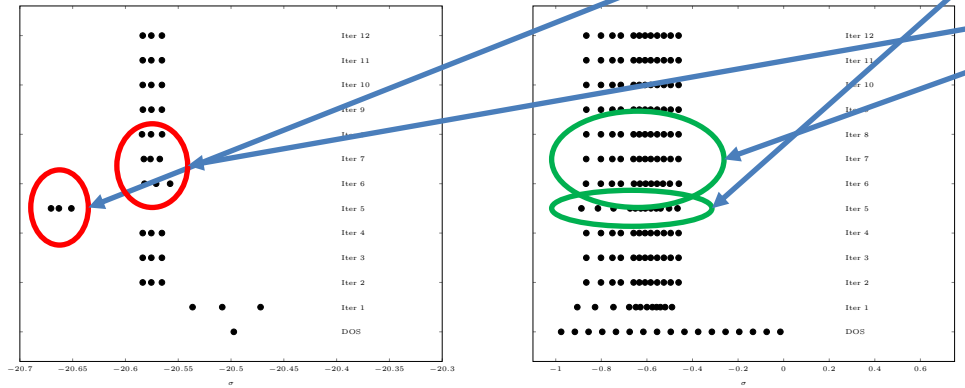


Eigenspectrum of $SiOSi_5$ Exhibits Multiple Distribution Regimes



SISLICE Exhibits Proper Convergence Behaviour in SCF



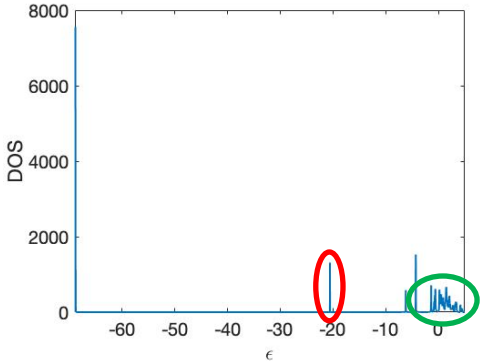


Non-smooth SCF convergence

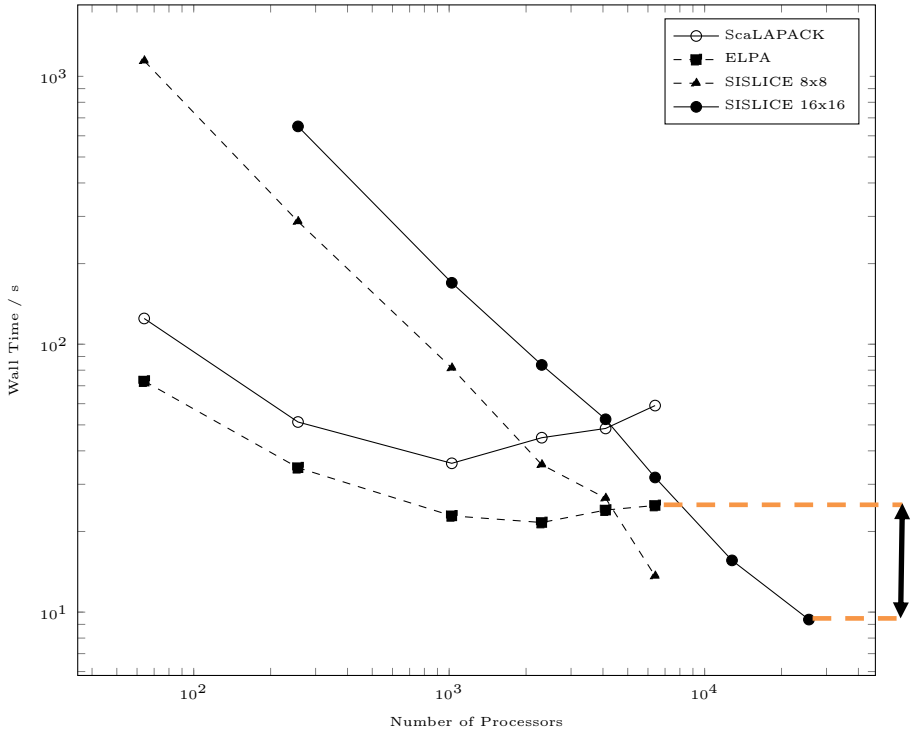
K-means shift migration

SCF recovery

Shift placement recovery



SISLICE Exhibits Linear Strong Scaling

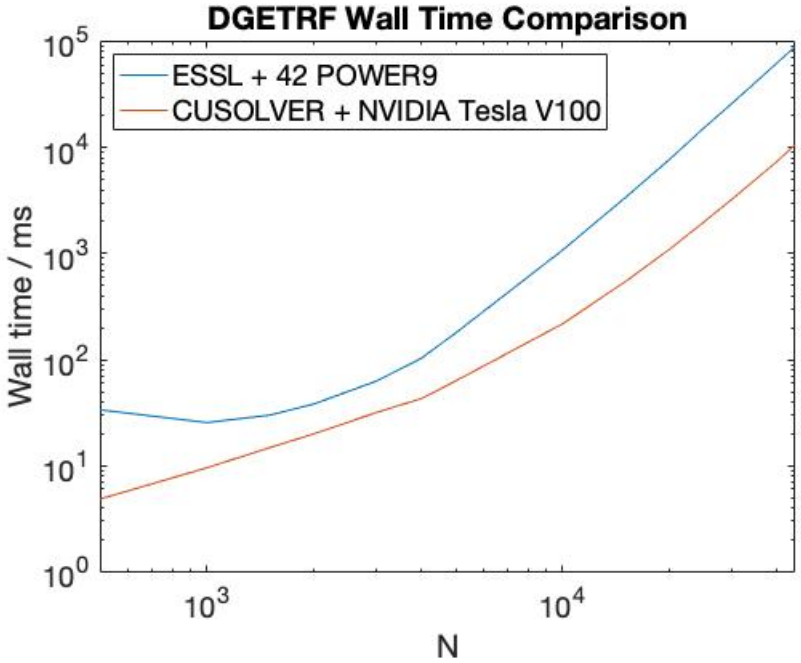


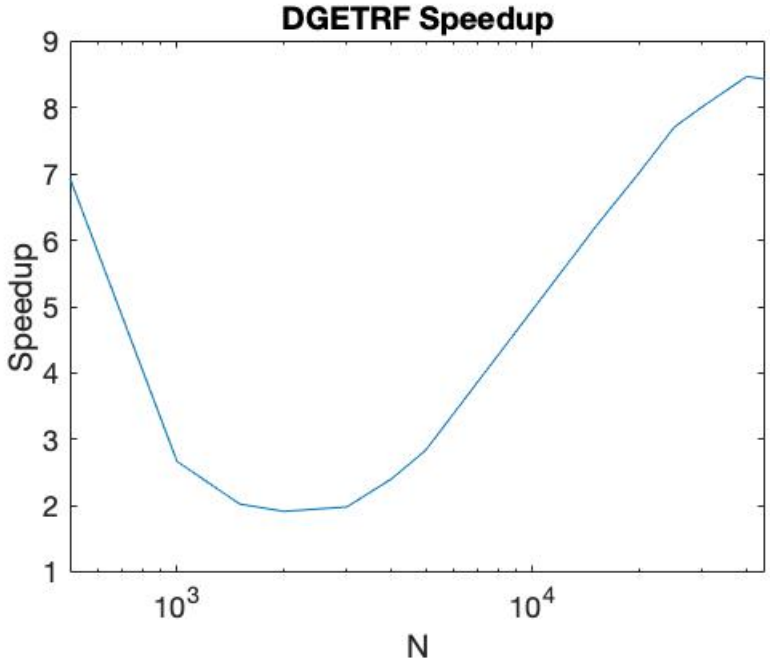
$Si_{10}H_{16}$ (UF Sparse Matrix Collection)

- N = 17077
- NNZ = 87592 (99.7% zeros)
- SuperLU for distributed LU factorization
- NB = 128 for ScaLAPACK/ELPA

2.7x speedup

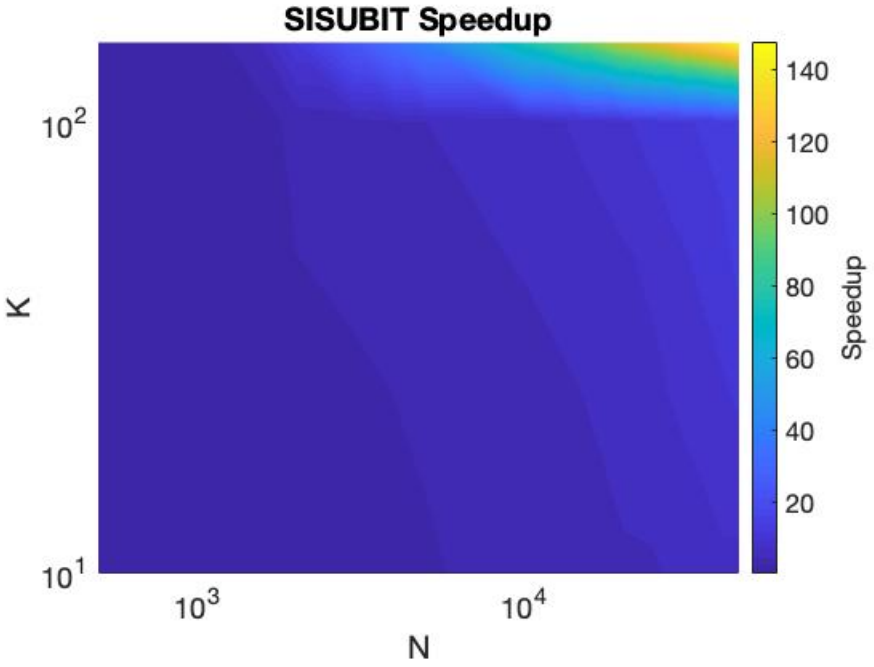
The Shift-Invert Subspace Iteration Performs Well on GPUs

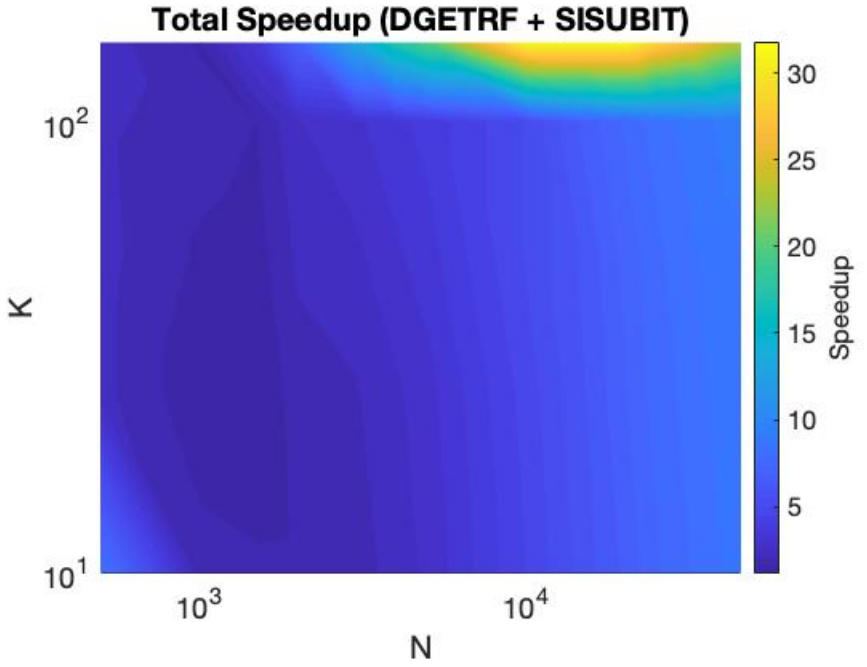




$$(F - \sigma S) \rightarrow LU$$

The Shift-Invert Subspace Iteration Performs Well on GPUs





$V \rightarrow U^{-1}L^{-1}V$ (8x)

Conclusions

- DOS estimation enables the expression of additional concurrency over other spectrum slicing implementations
- Clustering allows for tracking of the dynamic eigenvalues throughout the SCF
- The SISLICE method offers a robust and high-performance avenue for electronic structure simulations

Acknowledgements

The development of SISLICE has been supported by the U.S. Department of Energy:
- SciDAC-4
- Exascale Computing Project (NWChemEx 17-SC-20-SC)

Calculations were performed using DOE computing facilities:
- Cori (NERSC)
- Summit (OLCF)