Linear Scaling Solvers for Density Functional Theory Calculations

William Dawson
Takahito Nakajima

Computational Molecular Science Team
RIKEN Advanced Institute for Computational Science

Interdisciplinary Symposium on Modern Density Functional Theory

Standard Practice

- Choose a basis set $\{\boldsymbol{\phi}_u\}$ to represent the wavefunctions:
  - $\boldsymbol{\psi}_i = \sum c_{ui} \boldsymbol{\phi}_u$.
- Compute the overlap matrix:
  - $S_{uv} = \int \boldsymbol{\phi}_u^* \boldsymbol{\phi}_v$.
- Construct the operator matrix:
  - $H_{uv} = \int \boldsymbol{\phi}_u^* \mathcal{H} \boldsymbol{\phi}_v$.
- Solve the generalized eigenvalue problem:
  - $H_{uv} C = S_{uv} C\lambda$.
- Compute the density matrix in the chosen basis using the eigenvectors:
  - $P_{uv} = \sum c_{ui} c_{ui}$.
- However, this conventional method requires $O(N^3)$ operations.

$O(N^3)$ Is A Problem

- x10 the number of atoms requires x1000 the computational power.
- Large systems of interest: dilute solutions, defects in a solid, biological molecules, etc.
- Kohn's Nearsightedness principle: Changes of that potential, no matter how large, beyond a distance R have limited effects on local electronic properties.
- Practically speaking: H and P should be sparse, and only grow linearly with the system size.

Prodan, Emil, and Walter Kohn. "Nearsightedness of electronic matter." *Proceedings of the National Academy of Sciences of the United States of America* 102, no. 33 (2005): 11635-11638.

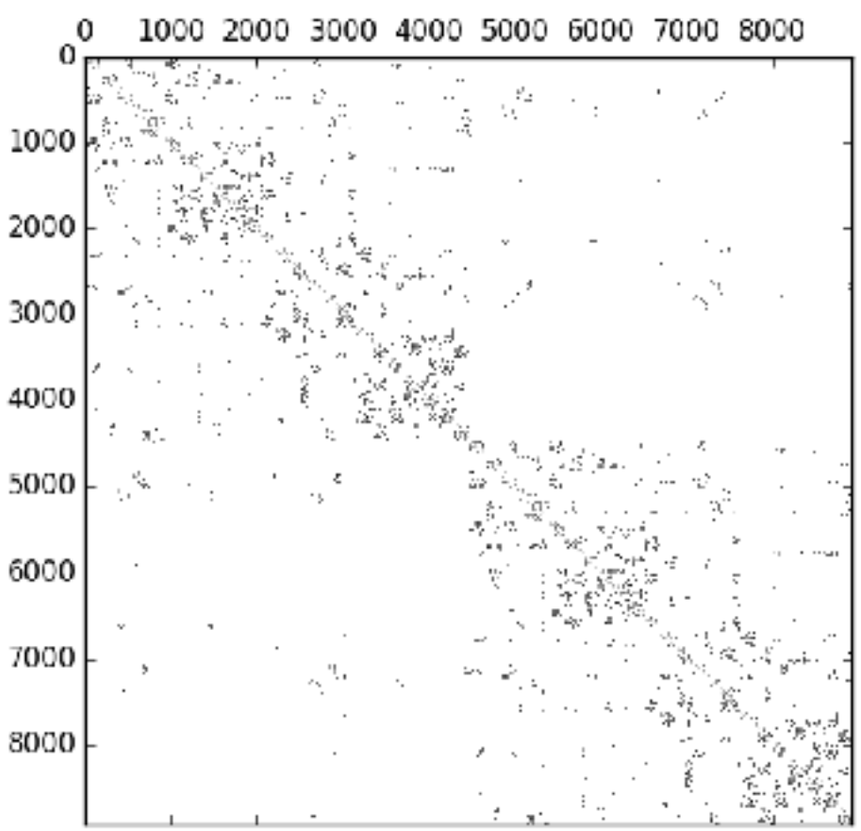
Locality Example

- Maximally Localized Wannier Functions (Boys Orbitals).
- Compute MLWF, and truncate outside of some radius.
- Example: 64 Water Molecules computed with a plane wave code.

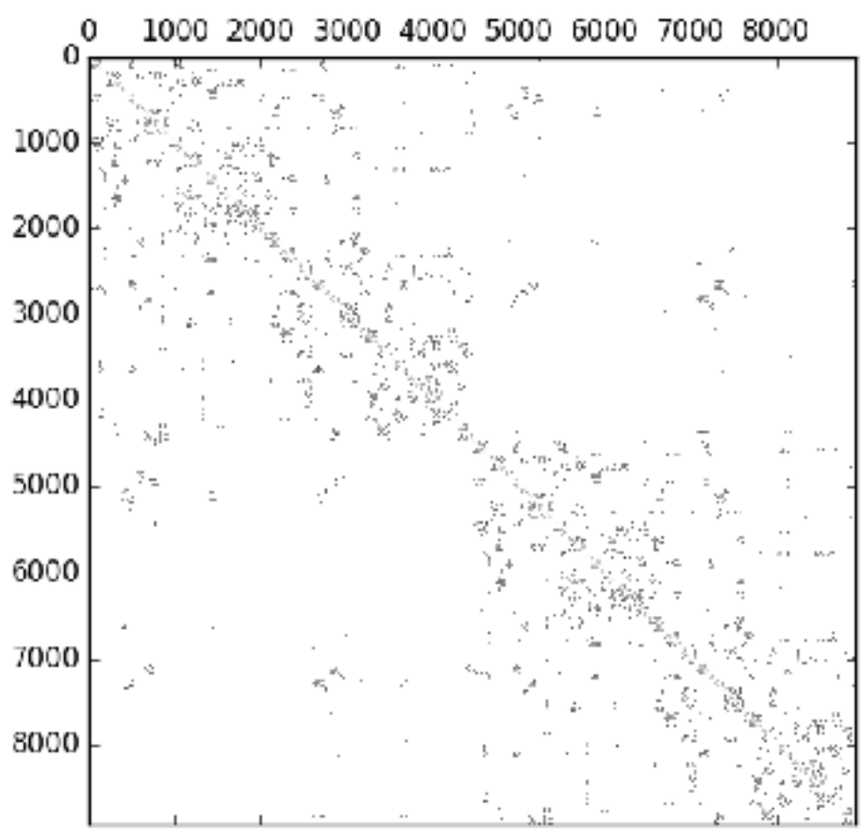
Radius (Bohr Radii)	Relative Energy Error
14	1.93E-06
10	3.46E-05
8	1.90E-04
5	2.52E-03

Locality Example-2

- CP2K linear scaling semi-empirical calculations.
- 2D polymer system, 3168 atoms.



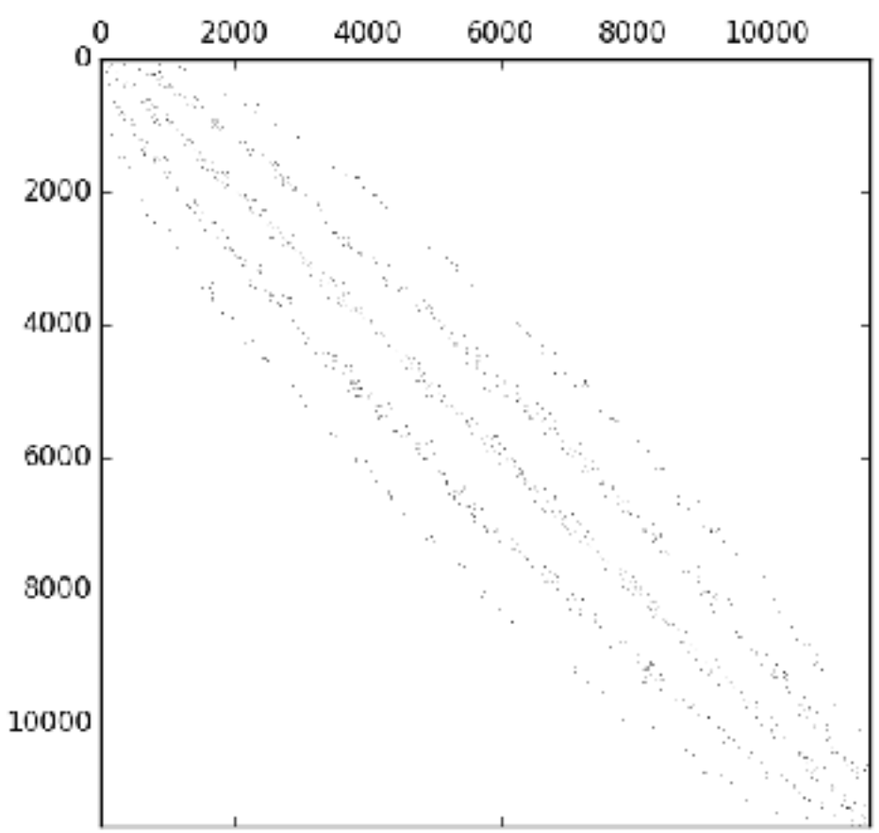
Hamiltonian



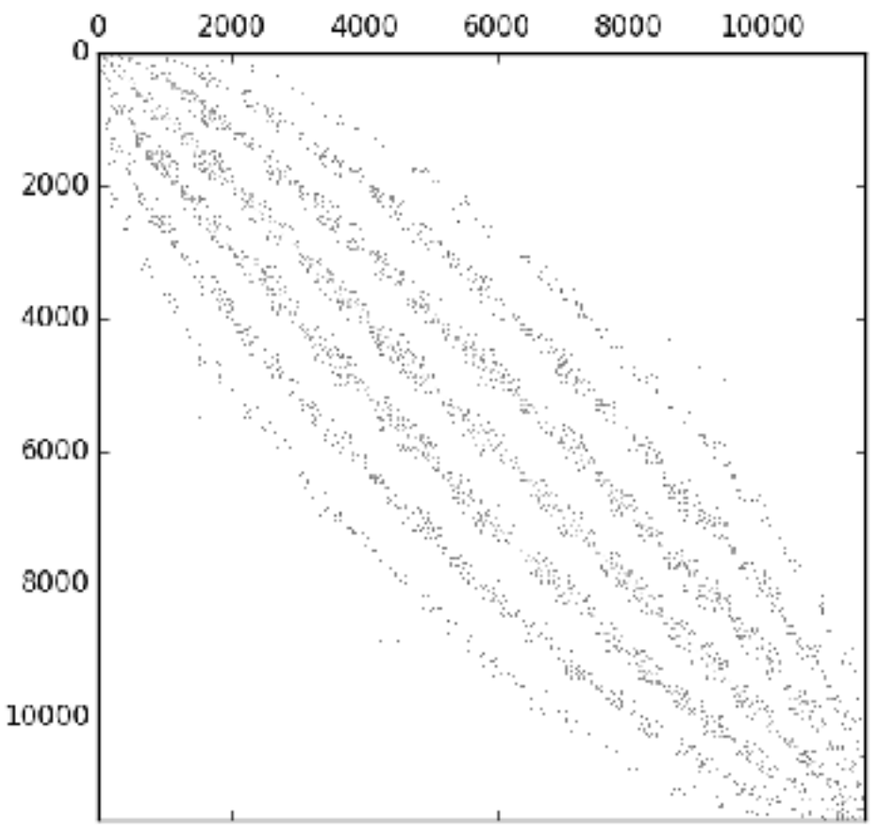
Density Matrix

Locality Example-3

- BigDFT linear scaling calculation.
- 1920 Water Molecule Cluster.



Hamiltonian



Density Matrix

Unified Framework For Solvers

- There exist a variety of linear scaling methods in the literature.
- For the dense eigenvalue problem, there exist a number of massively parallel libraries.
- For sparse eigenvalue problem as well, though aiming for dense eigenvectors.
- “No such standard library” for the sparse algorithms used in linear scaling quantum chemistry.
- Need a unifying mathematical framework that ties these variety of methods all together.

Matrix Functions

- Function of one variable: f(x).
- Function of a matrix: f(A).
- Examples:
  - $f(A) = A^2$.
  - $f(A) = A^{-1}$.
  - $f(A) = e^A$.
- Applications:
  - Differential Equations.
  - Graph Theory.
  - Lattice Quantum Chromodynamics.
  - Of course: Electronic Structure Calculation.

Higham, Nicholas J. *Functions of matrices: theory and computation*. Siam, 2008.

Matrix Function Applications

- Differential Equations:
  - $d^2y/dt^2 + Ay = 0$, $y(0) = y_0$, $y'(0) = y'_0$
  - $y(t) = \cos(\sqrt{A}t)y_0 + \sqrt{A}^{-1}\sin(\sqrt{A}t)y'_0$.
- Graph Theory:
  - Let A be the adjacency matrix.
  - The ease of sending a message from node i to j: e^{A}_{ij}.
  - The Katz centrality of a node: $(I-cA)^{-1}_{ii}$.

Estrada, Ernesto, and Naomichi Hatano. "Communicability in complex networks." *Physical Review E* 77, no. 3 (2008): 036111.

Katz, Leo. "A new status index derived from sociometric analysis." *Psychometrika* 18.1 (1953): 39-43.

Electronic Structure Applications-1

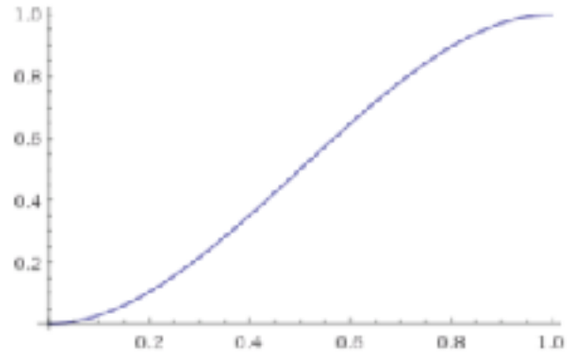
- Basis Set Orthogonalization (Lowdin):
  - Matrix inverse square root function.
  - $HC = SC\lambda$.
  - $S^{-1/2}HS^{-1/2}C = C\lambda$.
- Overlap Matrix Inversion (Orbital Minimization Method):

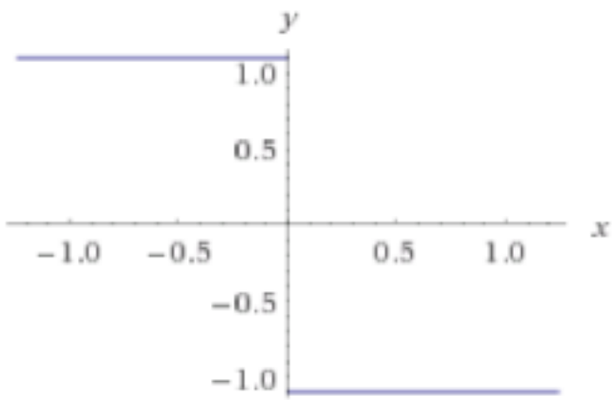
$$E_{KS}[\{\phi_i\}_{i=1}^N] = \sum_{i,j=1}^{N} \left(S^{-1}\right)_{ij} \int_\Omega \phi_i(\mathbf{r}) \left(-\frac{1}{2}\triangle\right) \phi_j(\mathbf{r}) d\mathbf{r}$$
$$+ \frac{1}{2}\int_\Omega \int_\Omega \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2 + E_{XC}[\rho]$$
$$+ \sum_{i,j=1}^{N} \left(S^{-1}\right)_{ij} \int_\Omega \phi_i(\mathbf{r}) \left[V_{ext}\phi_j\right](\mathbf{r}) d\mathbf{r},$$

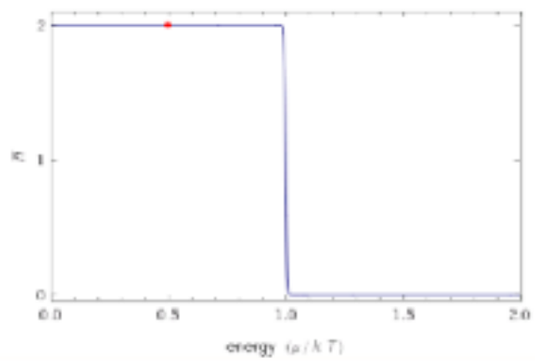
Fattebert, Jean-Luc, Daniel Osei-Kuffuor, Erik W. Draeger, Tadashi Ogitsu, and William D. Krauss. "Modeling dilute solutions using first-principles molecular dynamics: computing more than a million atoms with over a million cores." In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, p. 2. IEEE Press, 2016.

Electronic Structure Applications-2

- Curvy Step Density Matrix Minimization:
  - $P_{n+1} = e^{-X}P_n e^{X}$.
- McWeeny Purification:
  - $P_{n+1} = 3P_n^2 - 2P_n^3$.
- Matrix Sign Function:
  - $P = 0.5*(I - \text{sign}(H - \mu I))$.
- Fermi Operator Expansion:
  - $P = 0.5*(I-\text{erf}((H-\mu I)/(\Delta\varepsilon)))$.







How To Compute A Matrix Function

- If The Matrix Is Diagonalizable:
  - A = UDU*.
  - f(A) = Uf(D)U*.
  - Looks familiar…
- If It Is Not:
  - A variety of approaches, including Schur Decomposition based approach:
  - A = UTU*.
  - f(A) = Uf(T)U*. Explicit formulas for f(T) exists.

How To Compute A Matrix Function-2

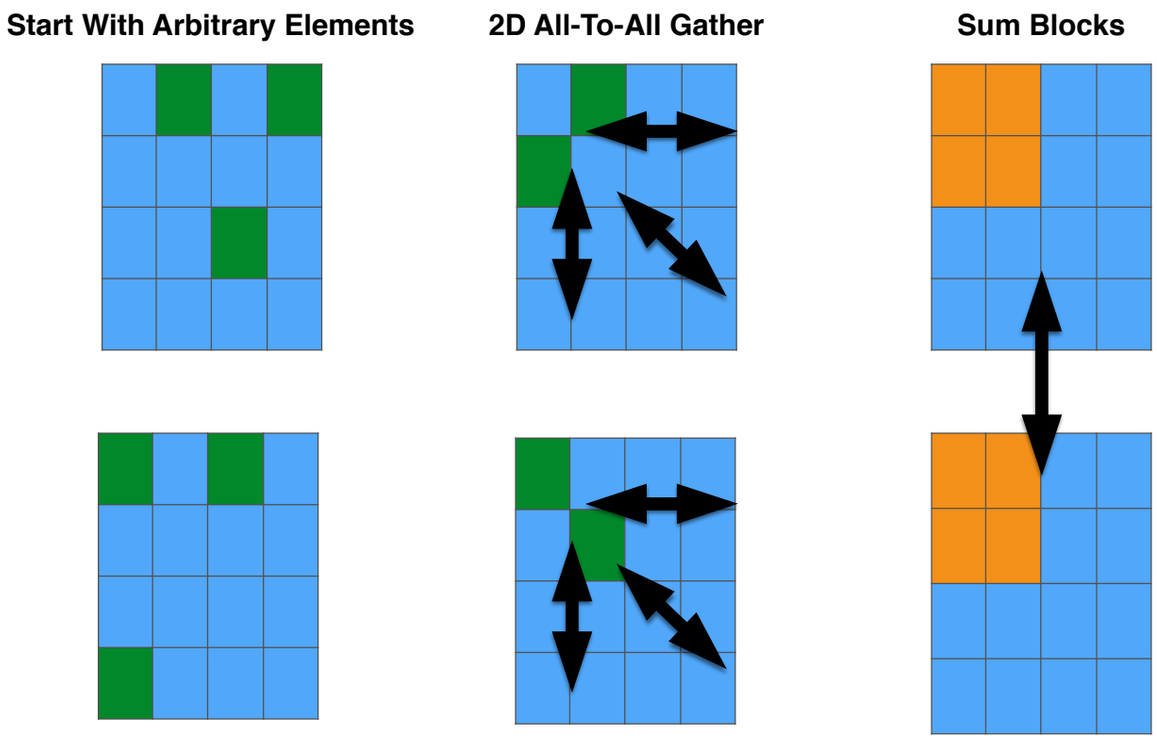
- Those direct methods are equally (or more) expensive then our original approach!
- Instead, we can approximate our function with polynomials based on Chebyshev Polynomials, truncated Taylor Series, etc…
- For example, Taylor series approximation of a matrix exponential:
  - $e^A = I + A + A^2/2! + A^3/3! + \ldots$
- Computing a matrix polynomial only requires matrix multiplication, addition.
- When the matrices are sparse, we can use sparse matrix multiplication and addition, which scale linearly with the system size.

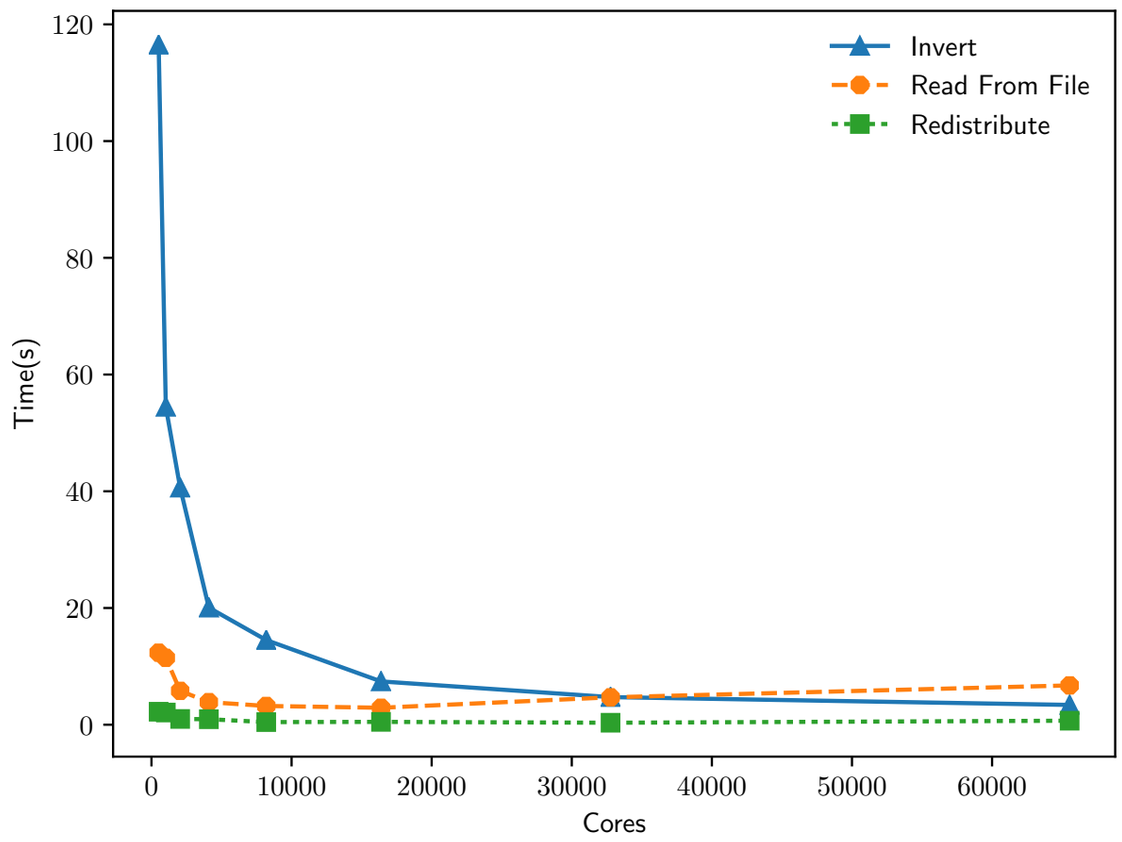
Feature Set Summary

- **General Polynomials**
  - Standard Polynomials
  - Chebyshev Polynomials
- **Transcendental Functions**
  - Trigonometric Functions
  - Exponential and Logarithm
- **Matrix Roots**
  - Square Root and Inverse Square Root
  - Arbitrary Rational Roots
- **Quantum Chemistry**
  - Density Matrix Minimization
  - Density Matrix Purification
  - Chemical Potential
- **Other**
  - Matrix Inverse
  - Sign Function
  - Load Balancing Matrices
  - File I/O

Library Integration - 1

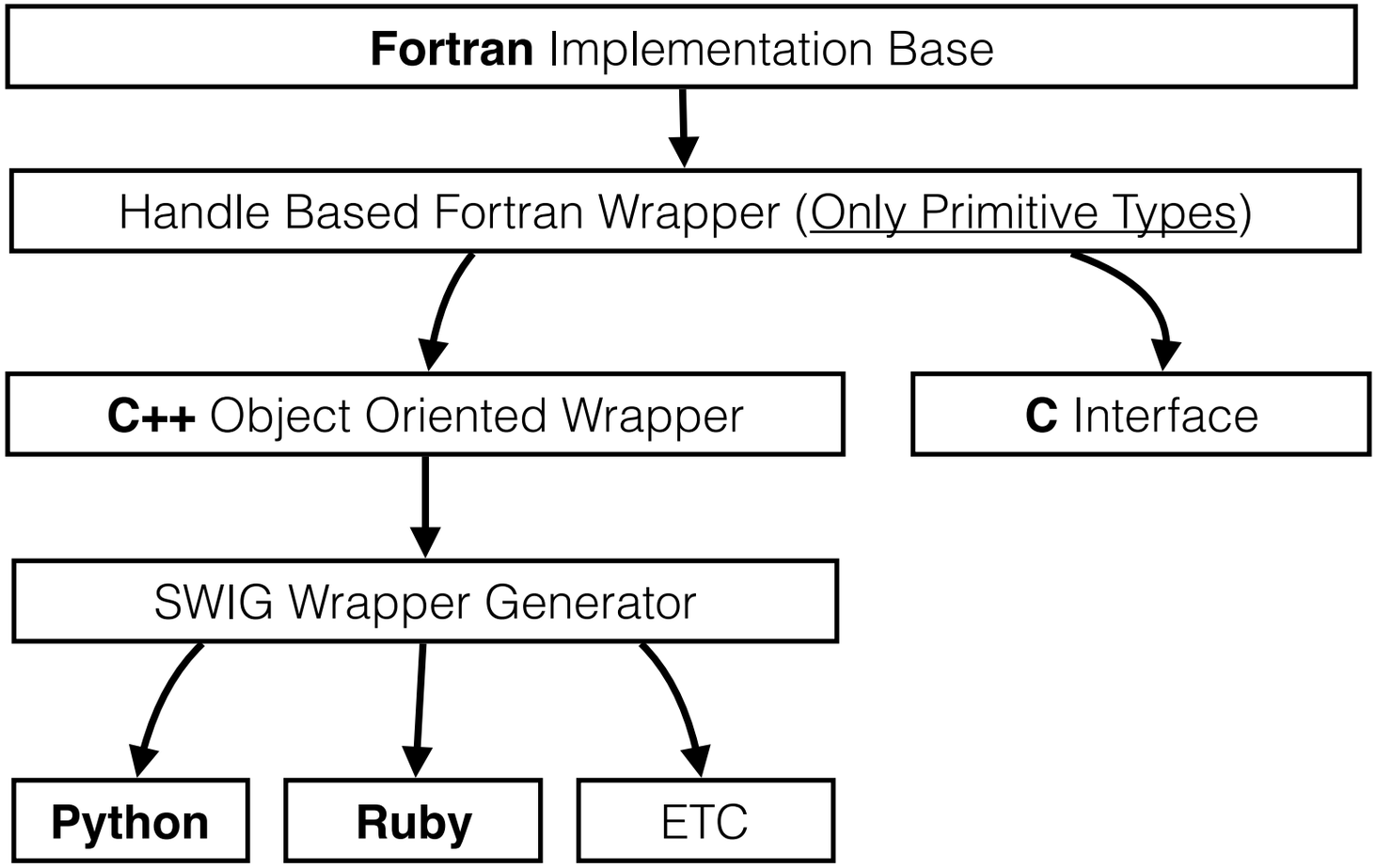
- Challenge: Integrate With Parallel Programs Using a Variety of Different Data Layouts.
- Solution 1: Parallel File I/O through the standard matrix market format for rapid prototyping.
- Solution 2: Arbitrary Data Remapping Routines.





Library Integration - 2

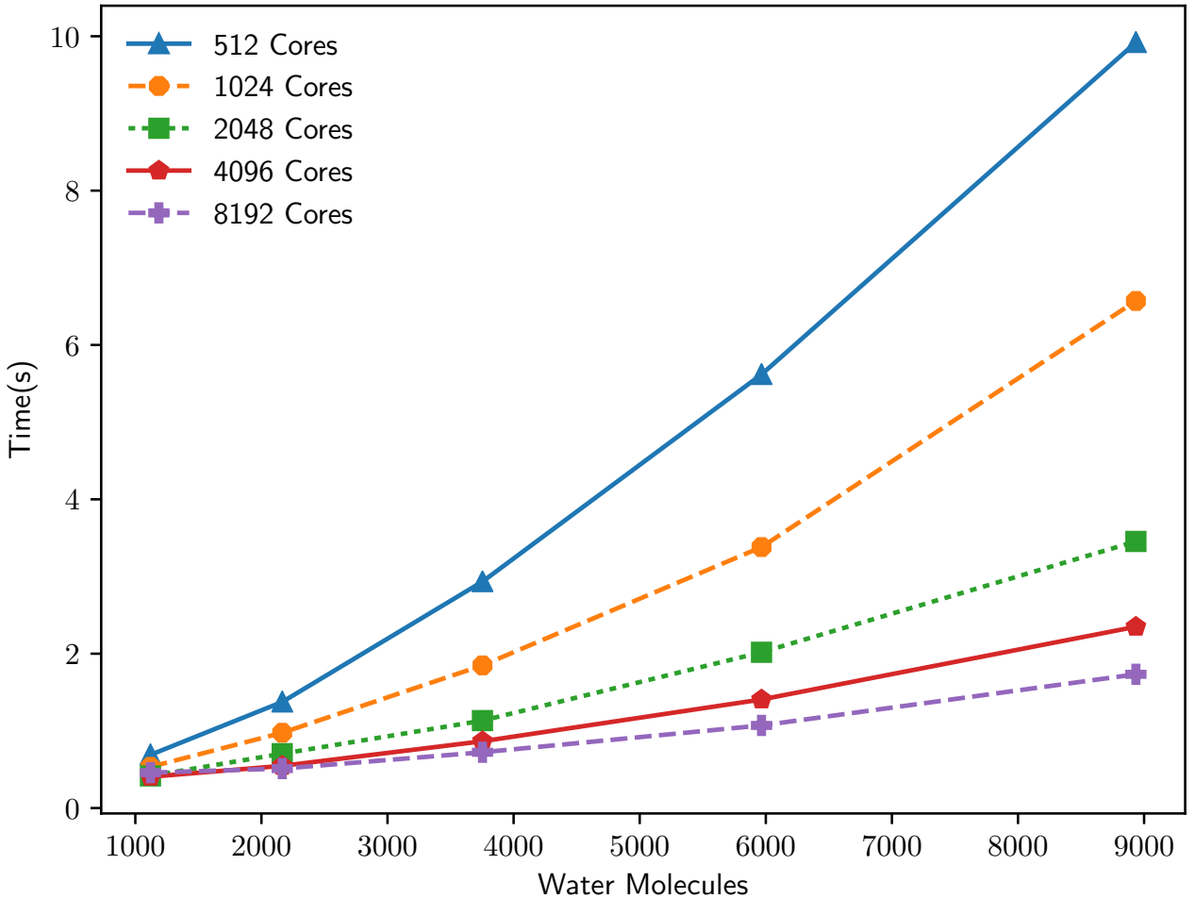
- Challenge: Integration with codes written in a variety of programming language.
- Solution: Programming Language Wrapper Hierarchy.



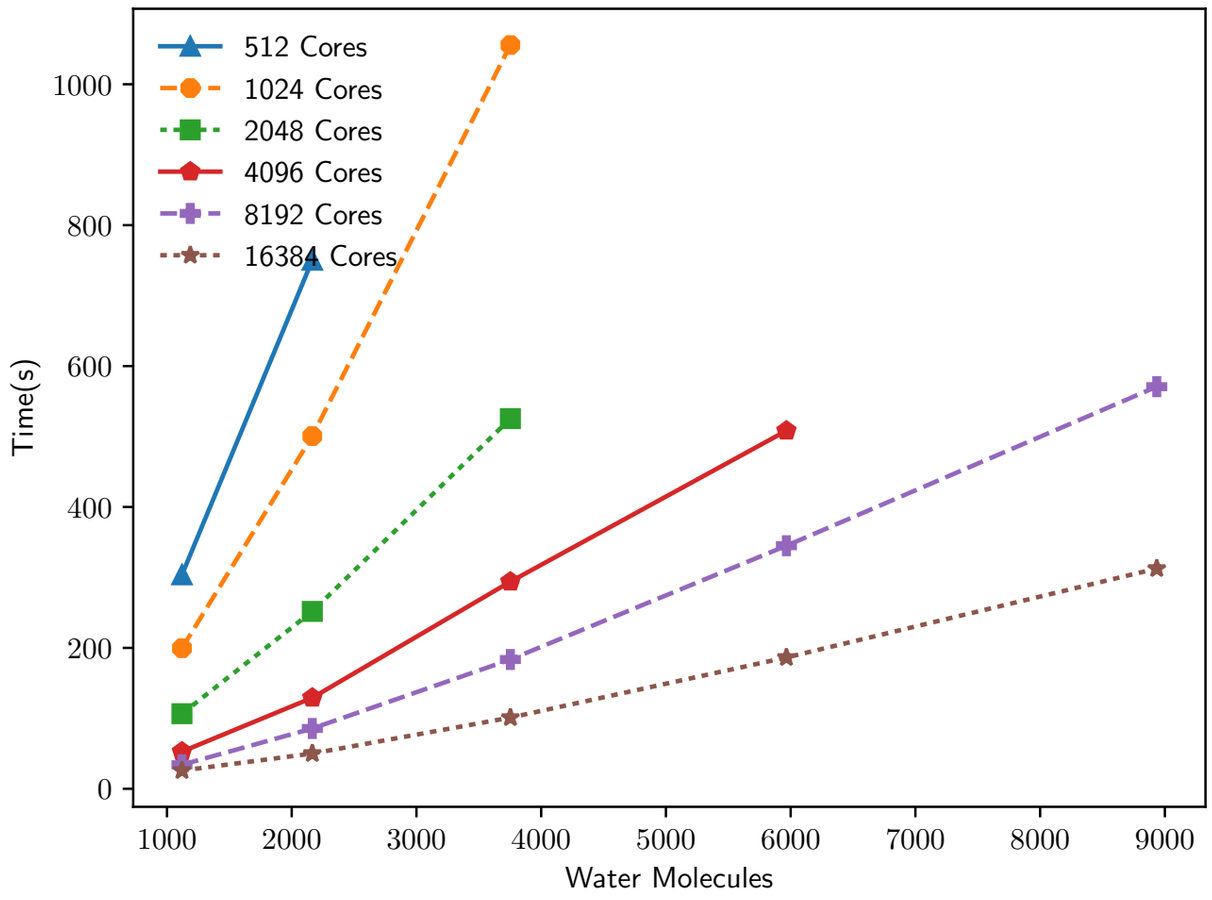
Application - 1

- Compute The Inverse Square Root of the Overlap Matrix.
- System: Water Clusters.

STO Basis Set



DZP Basis Set



Application - 2

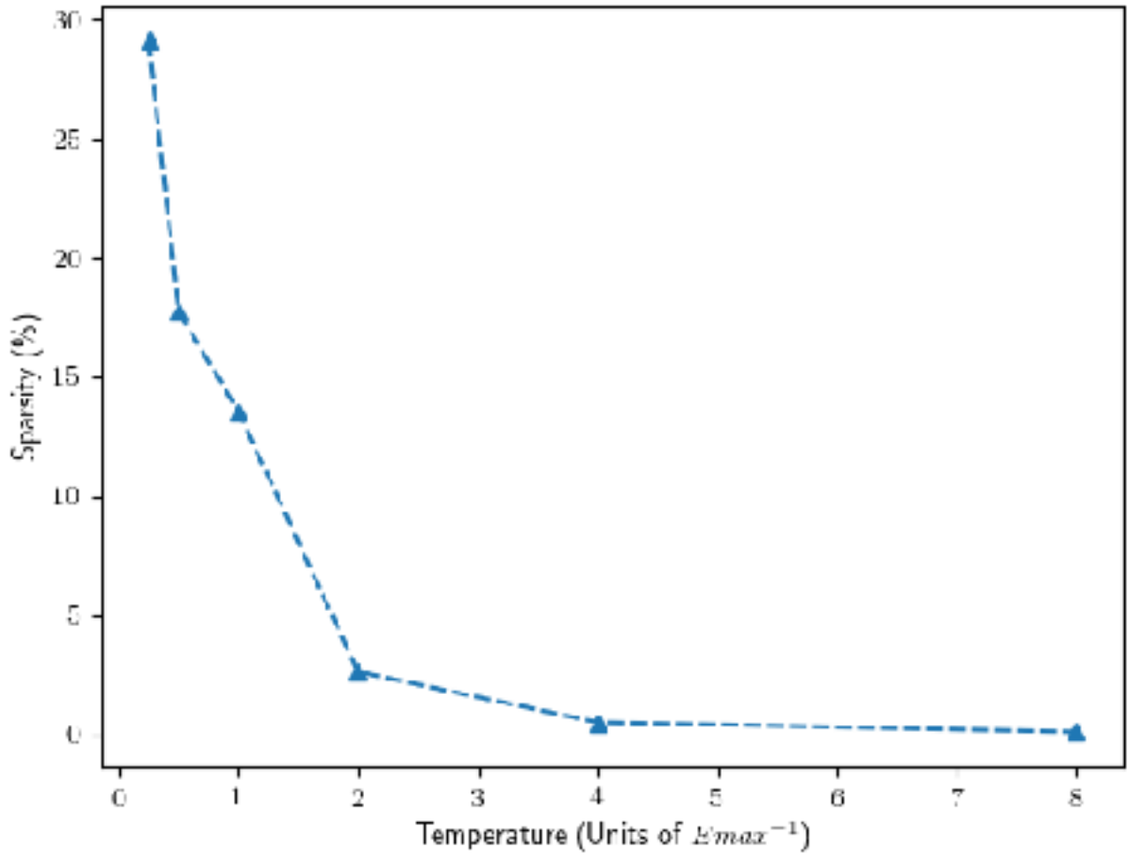
- Comparison of Density Matrix Solver Methods.
- Programs Compared: ErgoSCF, CP2K, BigDFT.
- Basis Sets: Gaussians, Gaussians, Wavelets.
- Theories: Hartree-Fock, Tight Binding, DFT (PBE).
- Solver Methods: TRS2, TRS4, HPCP.

	ErgoSCF	BigDFT	CP2K
Basis Functions	13,468	11,544	11,544
Sparsity of $S^{-\frac{1}{2}}HS^{-\frac{1}{2}}$	2.88%	1.36%	2.75%
Sparsity of P	7.36%	4.11%	5.85%
Occupation	71.4%	66.7%	66.7%

Method	ErgoSCF (s)	BigDFT (s)	CP2K (s)
TRS2	0.73	0.78	0.50
TRS4	0.71	0.78	0.43
HPCP	1.09	0.91	0.71

Application - 4

- Data Set: Israeli Social Network “TheMarker Cafe”
- Nodes: 69413. Sparsity: 0.04%.
- Artificially made symmetric.
- How does the ease of communicating with nearest neighbors affect the flow of information on the whole network?
- Estrada’s Scaled Matrix Exponential Metric: $e^{\beta A}$

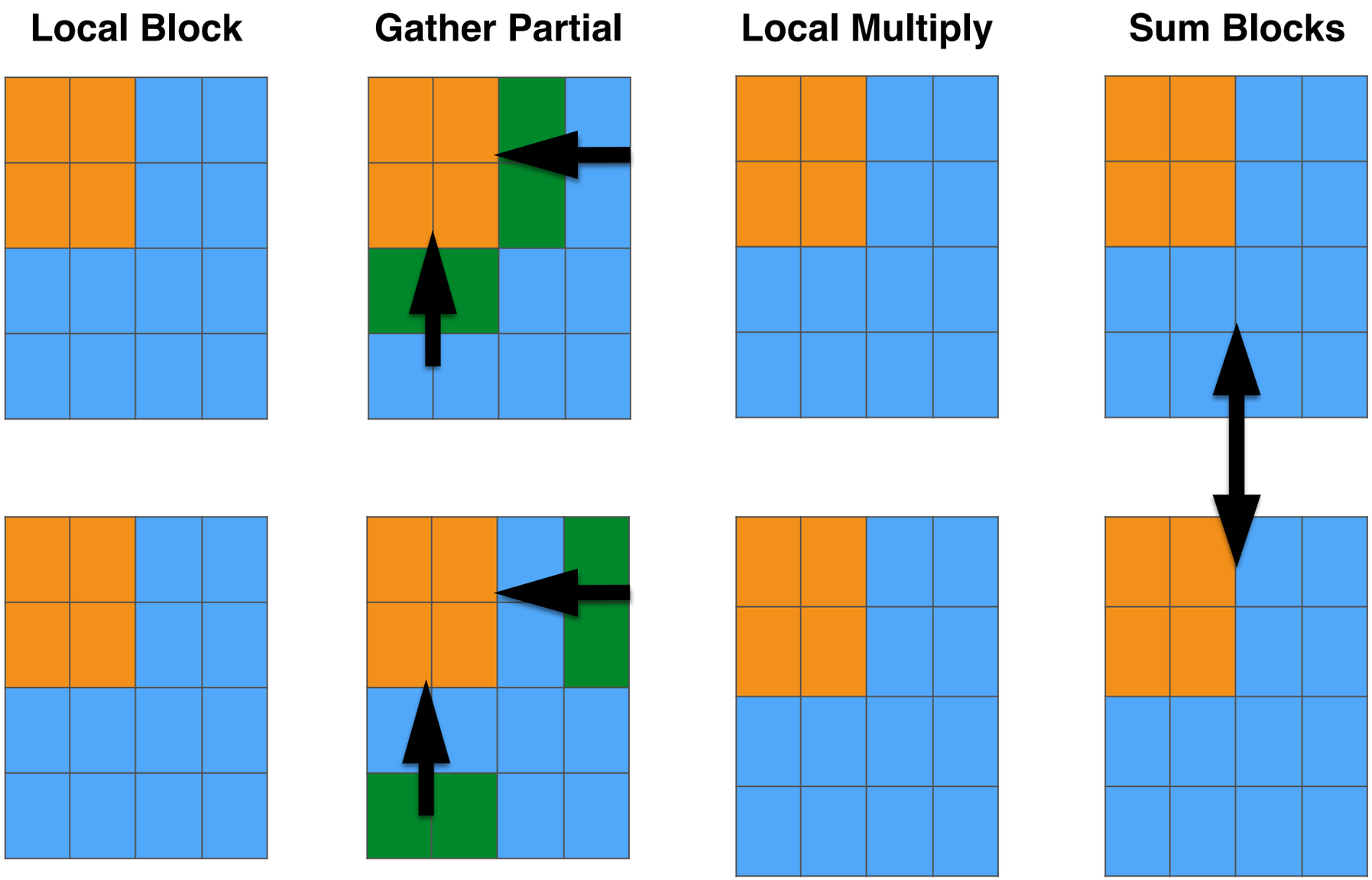


Conclusion

- Massively parallel library for computing the functions of sparse matrices.
- All functions are computed within one unified framework, and optimized by a single kernel (Sparse Matrix-Matrix Multiplication).
- Easy integration into a variety of different programs.
- Promising application areas well beyond quantum chemistry.
- Code availability: https://github.com/william-dawson/NTPoly

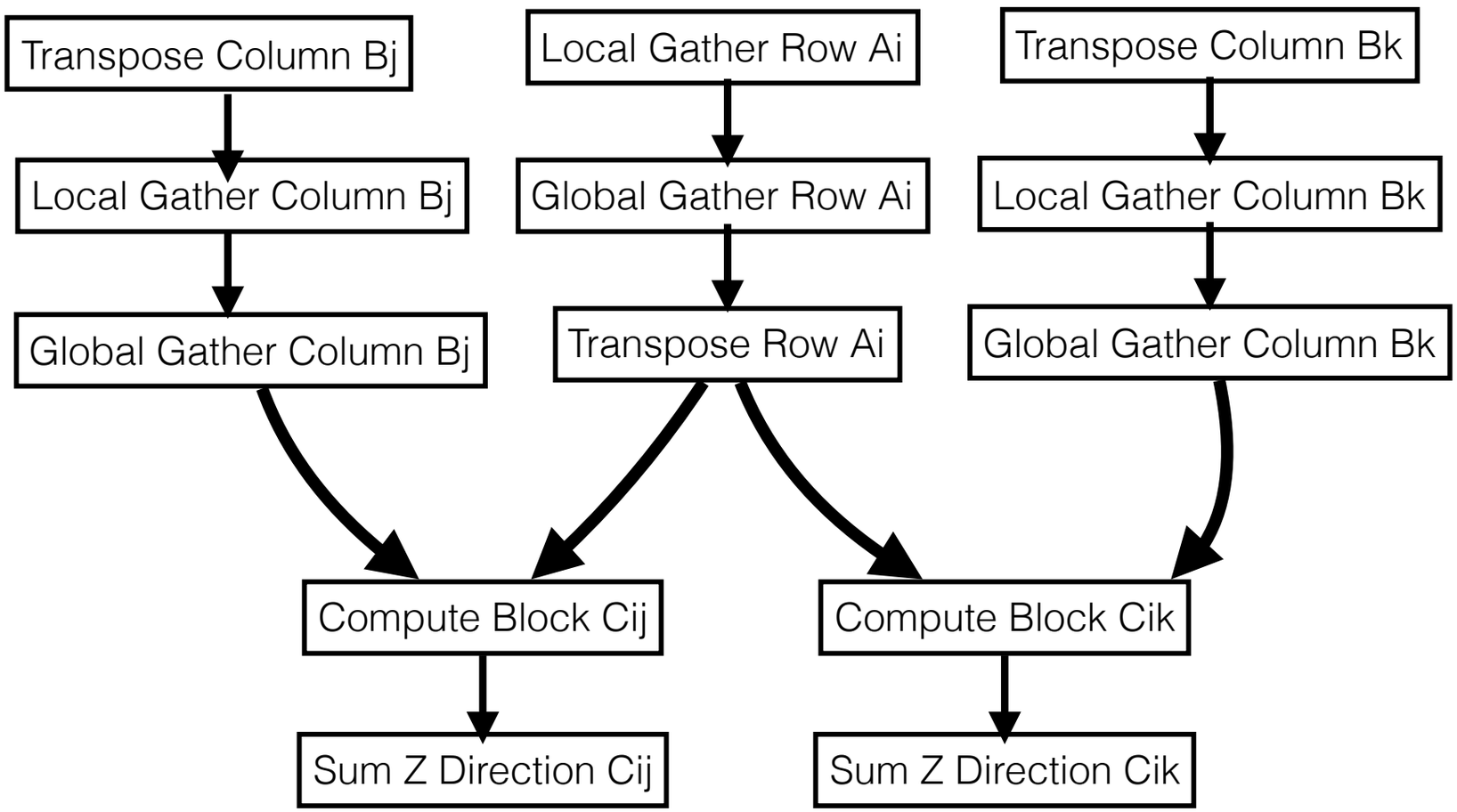
Bonus Slide - Distributed Matrix Multiply

- Based on a communication avoiding algorithm.



Bonus Slide - OpenMP Parallelism

- OpenMP Task Parallelism allows dynamic load balancing, communication/computation overlap.



Bonus Slide - Parallel Performance

- Compute the inverse of a nearly banded matrix.
- 32768 × 32768 matrix, 1% filled.