SLEPc-SIPs: Massively Parallel Sparse Eigensolver for Electronic Structure Calculations

Murat Keçeli
Computational Science Division
Argonne National Laboratory

August 16, 2018, MolSSI Workshop / ELSI Conference

Acknowledgment

Hong Zhang, Peter Zapol, Al Wagner, Âlvaro Vâzquez-Mayagoitia (ANL)

Jeff Hammond (Intel)

David Dixon (U of Alabama)

Jose Roman, Carmen Campos (Universitat Politècnica de València)

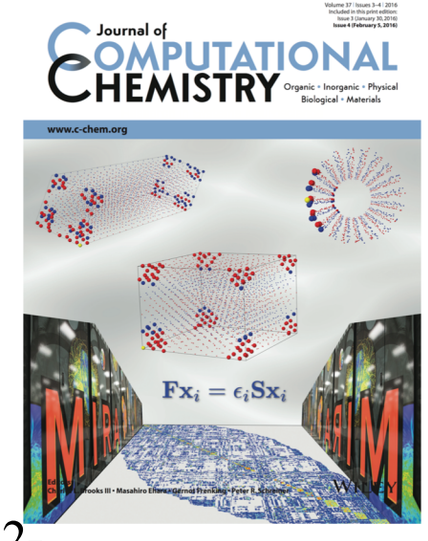
Fabiano Corsetti (Imperial College London)

Victor Yu, Volker Blum (Duke U)

Thanks to PETSc, SLEPc, Elemental, MUMPS, and PT-Scotch developers.

Work supported by US DOE, Office of Science under Contract No. DE-AC02-06CH11357.

Computations are done on Blues, Vesta, Mira, Theta at ANL, Cori at NERSC.



1. Zhang, H., Smith, B., Sternberg, M. & Zapol, P. SIPs. *ACM Trans. Math. Softw*. 33, 9 (2007).
2. Campos, C. & Román, *J. E. Numer. Algorithms*, 60, 279 (2012).
3. Keçeli M., Zhang, H., Zapol, P., Dixon A.D., & Wagner A. F., *J. Comput. Chem*. 37, 448 (2016).
4. Keçeli M., Corsetti F., Campos C., Roman J., Vâzquez-Mayagoitia A, Zhang, H., Zapol, P., & Wagner A, F. *J. Comput. Chem*. (2018).

The eigenvalue problem $\mathbf{Fx} = \lambda \mathbf{Sx}$

$$\begin{pmatrix} F_{11} & & & \\ F_{21} & F_{22} & & \\ \vdots & \vdots & \ddots & \\ F_{N1} & F_{N2} & \cdots & F_{NN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \lambda \begin{pmatrix} S_{11} & & & \\ S_{21} & S_{22} & & \\ \vdots & \vdots & \ddots & \\ S_{N1} & S_{N2} & \cdots & S_{NN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

- Can be the scaling bottleneck of HF and DFT based methods.
- 20% - 80% of the eigensolutions might be required.
- Eigenvalue spectrum can be clustered and may contain large gaps.
- Problem size scaling is cubic for flops, quadratic for memory.
- Matrices are naturally sparse for *large* problems with localized basis sets.

Eigensolvers

- Matrix representation:

Dense
- ❑ Uniform data layout, faster access
- ❑ Memory **O(N^2)**
- ❑ Computation **O(N^3)**
- ❑ ScaLAPACK, Elemental, ELPA

Sparse
- ❑ Nonuniform data layout, slower access
- ❑ Memory **O(N) - O(N^2)**
- ❑ Computation **O(N) - O(N^3)**
- ❑ SLEPc, Pardiso, MUMPS

- Solution algorithm

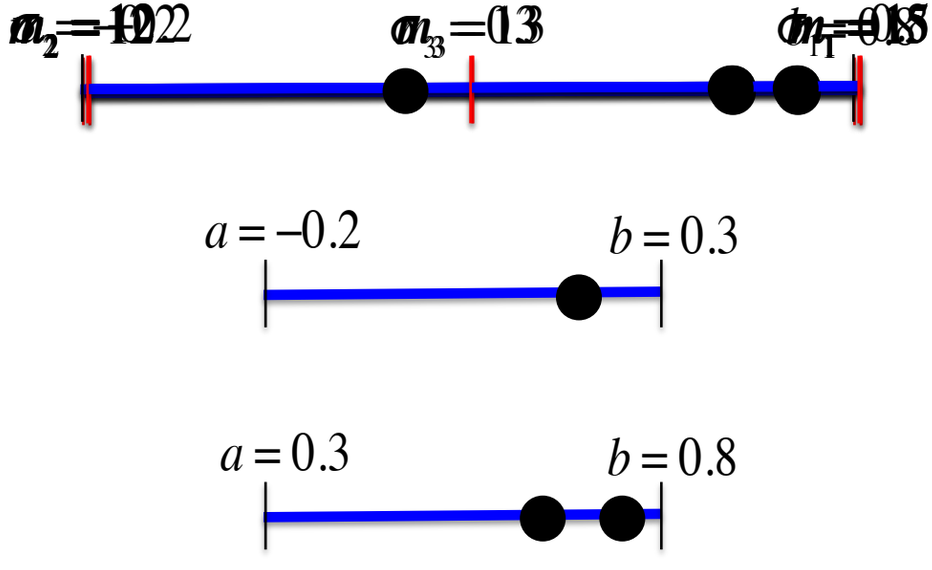
Direct
- ❑ Results in finite amount of steps
- ❑ Based on transformations or factorizations.
- ❑ For large portion of eigensolutions
- ❑ Robust, generally applicable
- ❑ Nonzero structure may change

Iterative
- ❑ Number of steps depends on input
- ❑ Based on initial guess
- ❑ For small portion of eigensolutions
- ❑ Accuracy depends on input
- ❑ Nonzeros structure is preserved

SIPs: Shift-and-Invert Parallel Spectral Transformations

Find all the eigenpairs in a given interval [*a*, *b*].

1. Shift: $\left(\mathbf{F}-\sigma_i\mathbf{S}\right)\mathbf{x}_i=\left(\lambda_i-\sigma_i\right)\mathbf{S}\mathbf{x}_i$
2. Factorize: $\mathbf{F}-\sigma_i\mathbf{S}=\mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}}$
3. Compute inertia: m_i
4. Invert: $\mathrm{K}=\left(\mathbf{F}-\sigma_1\mathbf{S}\right)^{-1}\mathbf{S}$
5. Solve: $\mathbf{K}\mathbf{x}_i=\dfrac{1}{\lambda-\sigma_1}\mathbf{x}_i$
6. Repeat 1-5 if necessary



1. Zhang, H., Smith, B., Sternberg, M. & Zapol, P. SIPs. ACM Trans. Math. Softw. 33, 9–es (2007).
2. Keçeli M., Zhang, H., Zapol, P., Dixon A.D., Wagner A, J. Comput. Chem. 37, 448 (2016) .

Parallelization

- Two layers of parallelism based on MPI.
- Horizontal one through slicing. (no communications)
- Vertical one through in slice operations. (heavy communications)

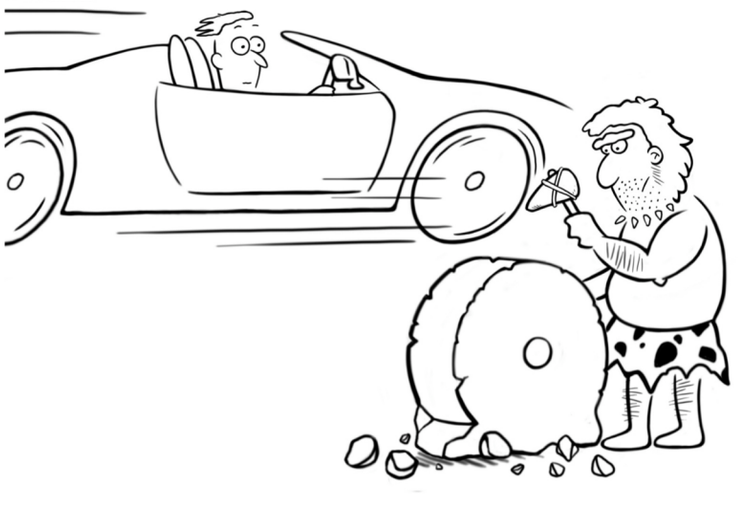
slice 1	slice 2	…	…	…	…	…	slice 8
rank=0 idEps=0 idMat=0	rank=4 idEps=1 idMat=0	…	…	…	…	…	rank=28 idEps=7 idMat=0
rank=1 idEps=0 idMat=1	rank=5 idEps=1 idMat=1	…	…	…	…	…	rank=29 idEps=7 idMat=1
rank=2 idEps=0 idMat=2	rank=6 idEps=1 idMat=2	…	…	…	…	…	rank=30 idEps=7 idMat=2
rank=3 idEps=0 idMat=3	rank=7 idEps=1 idMat=3	…	…	…	…	…	rank=31 idEps=7 idMat=3

SIPs Implementation

Remember the parallel platform paradox

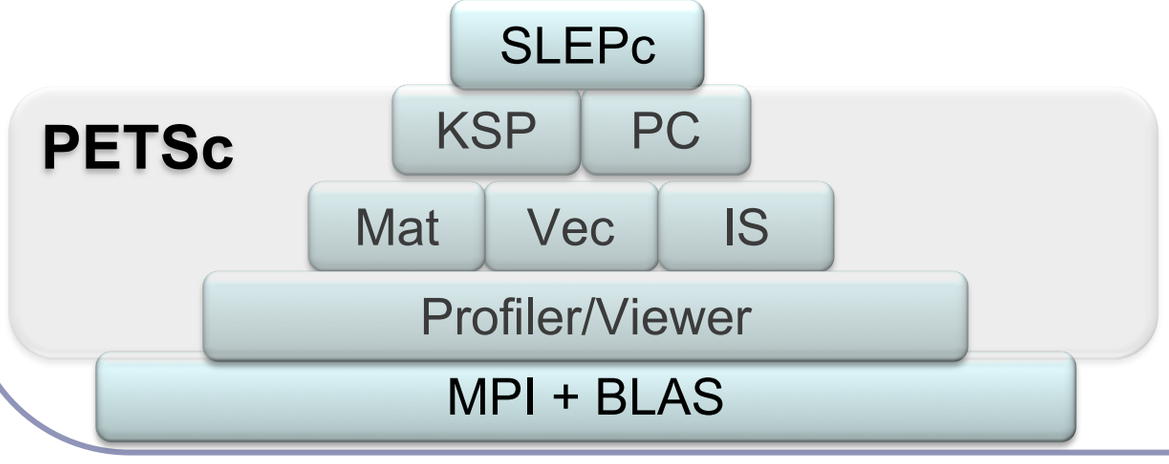
"The average time required to implement a moderate sized application is equivalent to half-life of the parallel computing platform", John Reynders, 1996.

- Use well-designed mathematical libraries as the building blocks.



Building Blocks

- PETSc: https://www.mcs.anl.gov/petsc/
  - Portable, Extensible Toolkit for Scientific Computation
  - A suite of data structures and routines for the scalable (parallel) solution of scientific applications modeled by partial differential equations.
  - 2009 R&D Award, 2015 SIAM/ACM prize, 5 Gordon Bell prizes for PETSc applications
- SLEPc: http://slepc.upv.es/
  - Scalable Library for Eigenvalue Problem Computations
  - Built on top of PETSc



SVD Solver		
Cross Product	Cyclic Matrix	Thick R. Lanczos

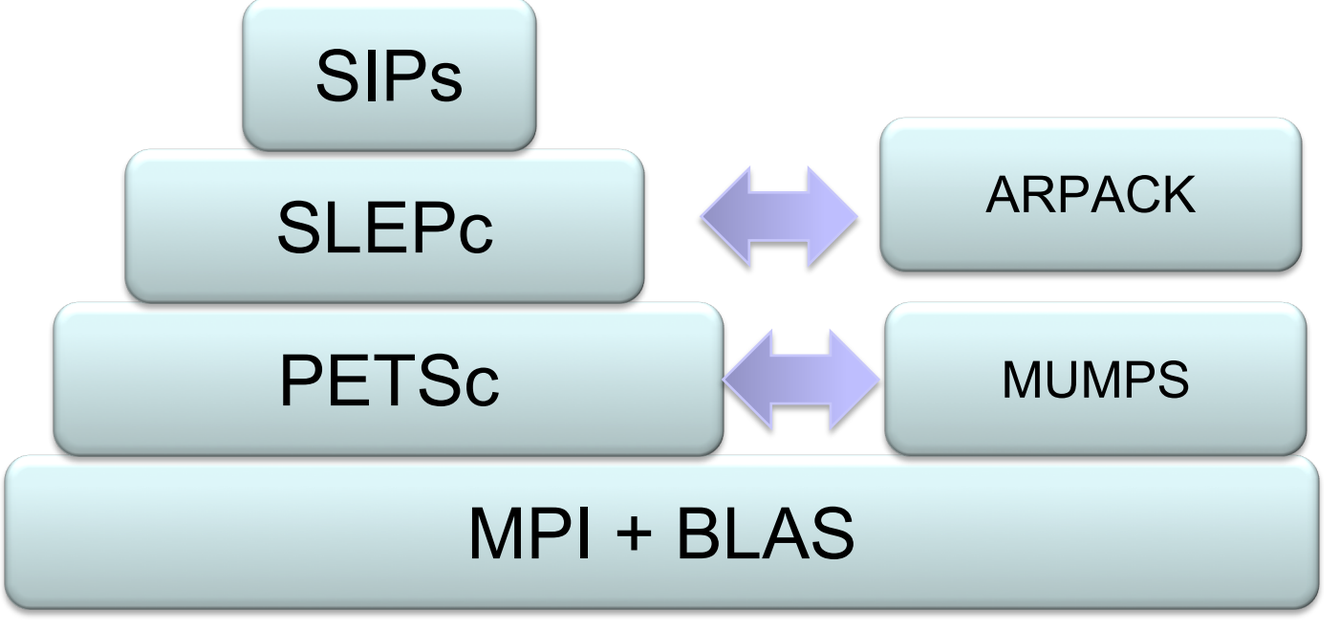
Polynomial Eigensolver		
TOAR	Linear-ization	Q-Arnoldi

Linear Eigensolver						
Krylov-Schur	Arnoldi	Lanczos	GD	JD	RQCG	CISS

Spectral Transformation			
Shift	Shift-and-invert	Cayley	Preconditioner

SIPs Implementation in 2007

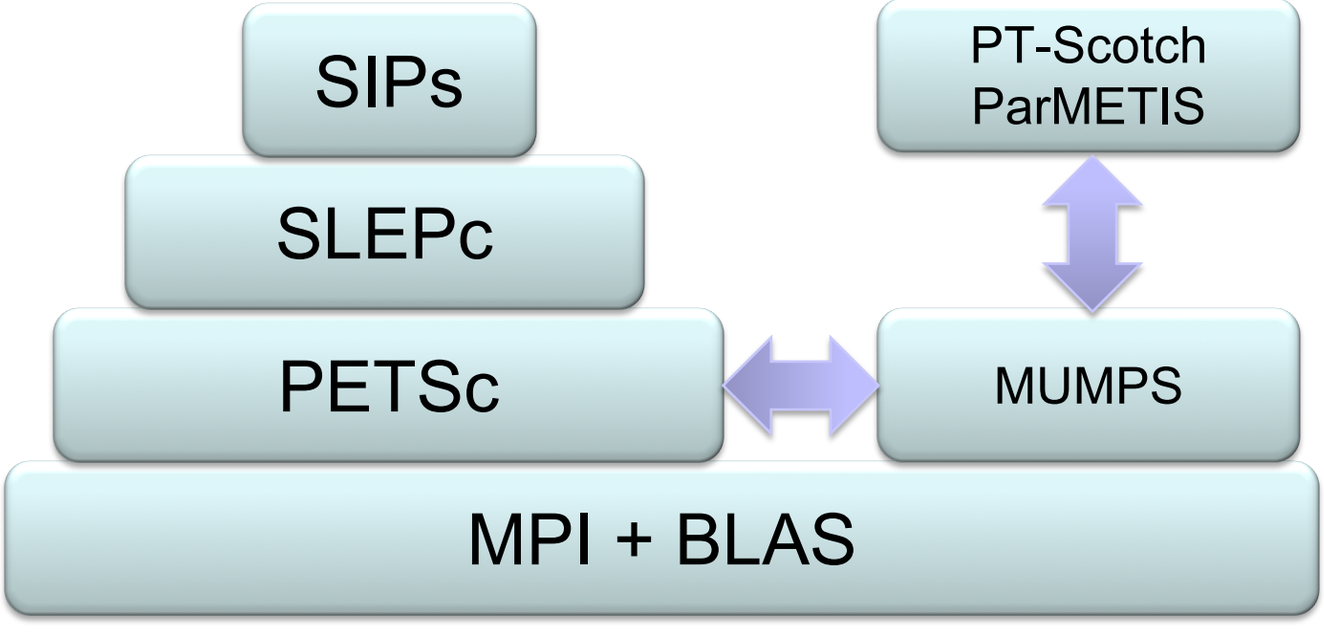
- Organize subgroups of MPI communicators.
- Select shifts
- Bookkeep and validate eigensolutions
- Balance parallel workload



1. Zhang, H., Smith, B., Sternberg, M. & Zapol, P. SIPs. *ACM Trans. Math. Softw*. 33, 9 (2007).

SIPs Implementation in 2014

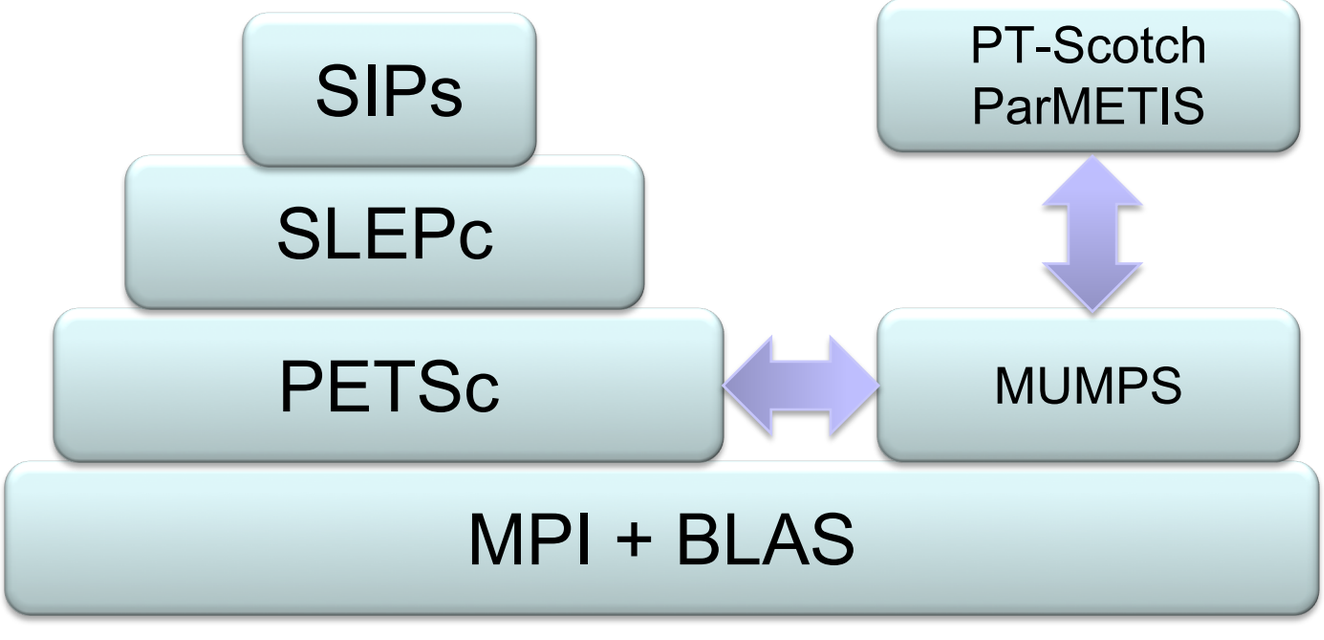
- Organize subgroups of MPI communicators.
- Select slices (uniform width)
- Bookkeep eigensolutions
- Use parallel matrix reordering



1. Campos, C. & Román, *J. E. Numer. Algorithms*, 60, 279 (2012).
2. Keçeli M., Zhang, H., Zapol, P., Dixon A.D., Wagner A, J. Comput. Chem. 37, 448 (2016) .

SIPs Implementation in 2016

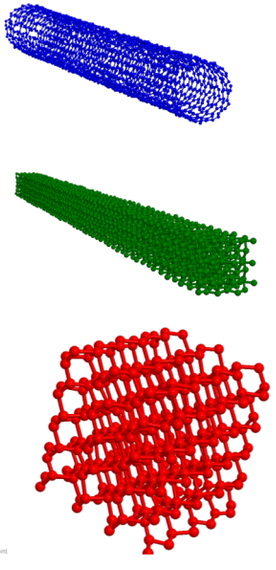
- Adjust slices with inertia or eigenvalue information
- Compute density matrix
- SIPs → QETSc → SIPs → SLEPc-SIPs



1. Keçeli M., Corsetti F., Campos C., Roman J., Vâzquez-Mayagoitia A, Zhang, H., Zapol, P., & Wagner A, F. *J. Comput. Chem*. (2018)

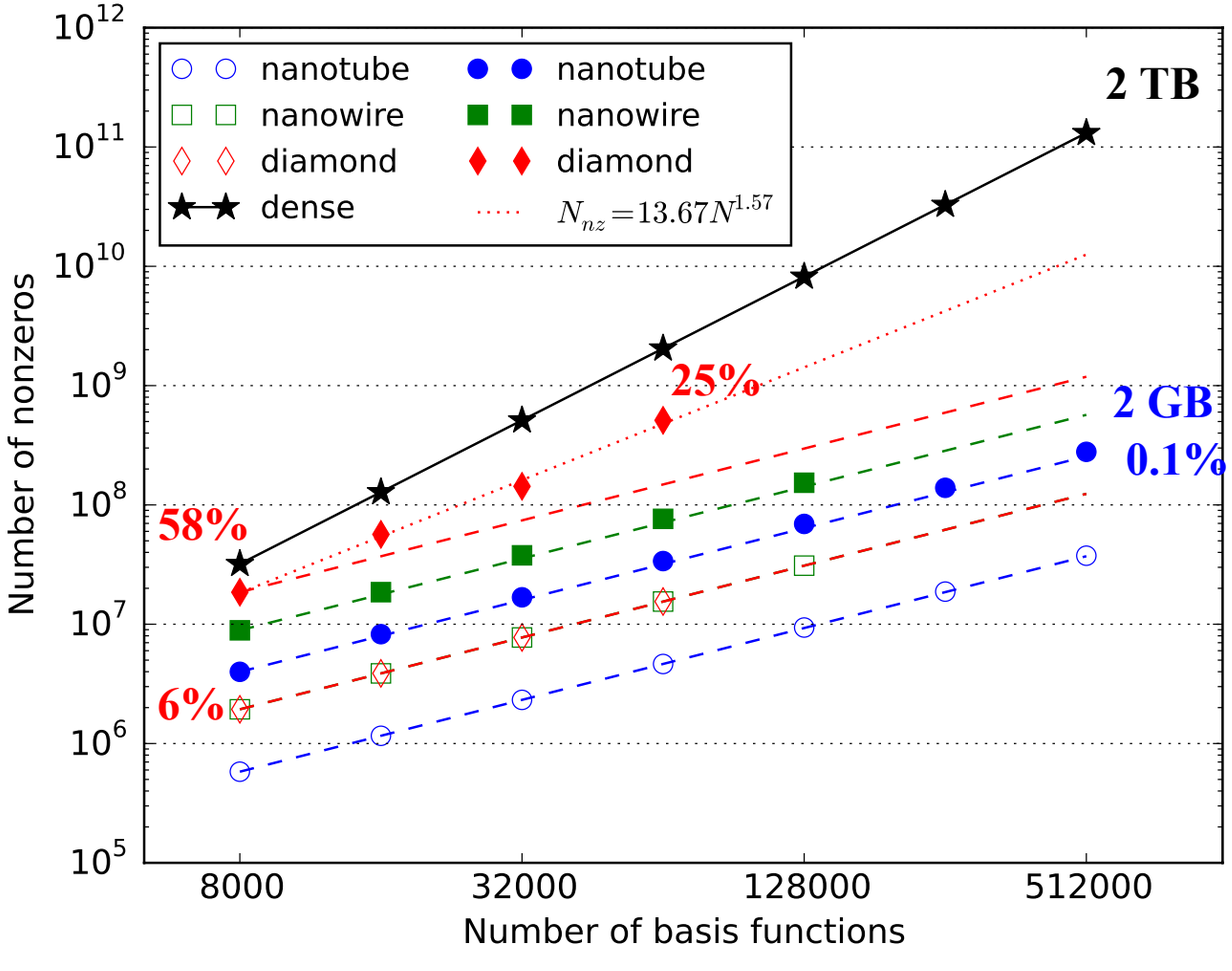
Benchmark calculations

- Matrices from DFTB gamma point calculations.
- Carbon NanoTube: CNT8000 – CNT512000
- Diamond NanoWire: DNW8000 – DNW128000
- Bulk Diamond Crystal: BDC8000 – BDC64000
- Positions of atoms are randomly deviated from eqb.
- More than 60% of eigenpairs are computed.
- All calculations are done on ALCF supercomputers
  - 786,432 IBM BG/Q cores
  - 16 cores per node,
  - 1 GB RAM per core
  - Peak at 10 petaflops
  - Interconnect: 5D Torus



Argonne NATIONAL LABORATORY

Sparsity after factorization



Filled symbols show the number of nonzeros after factorization

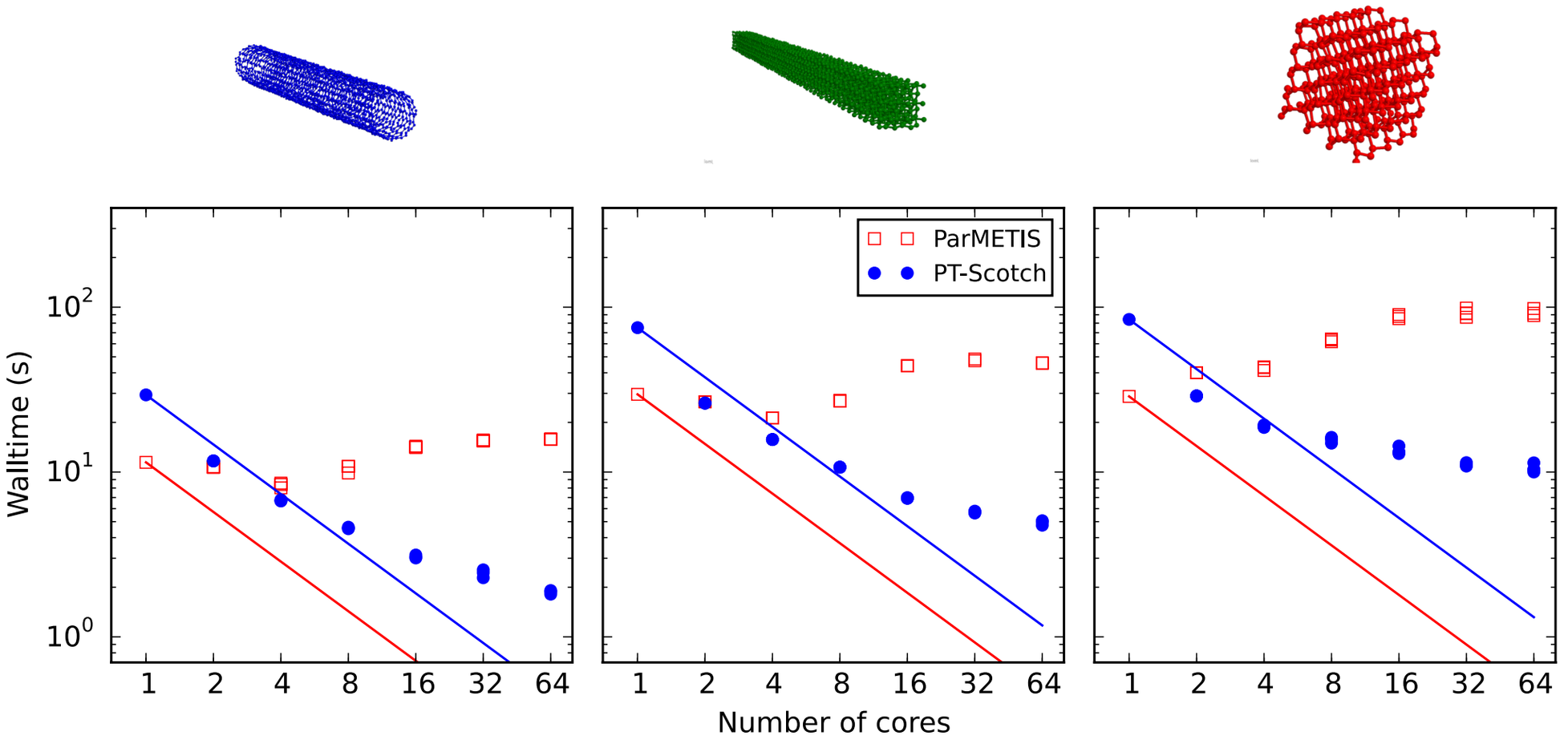
Reordering methods

- Fill-in ratio is the nnz of the factor divided by the nnz of the original matrix.
- Different reordering algorithms give different fill-in ratios.

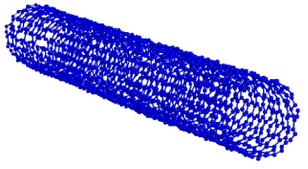
Type[a]	*RCM*	*AMD*	*AMF*	***METIS***	***Scotch***
CNT8000	7.7	8.0	7.7	**6.9**	**6.9**
CNW8000	5.4	5.7	4.9	**4.8**	**4.6**
BDC8000	11.1	11.2	10.5	**9.6**	**9.5**

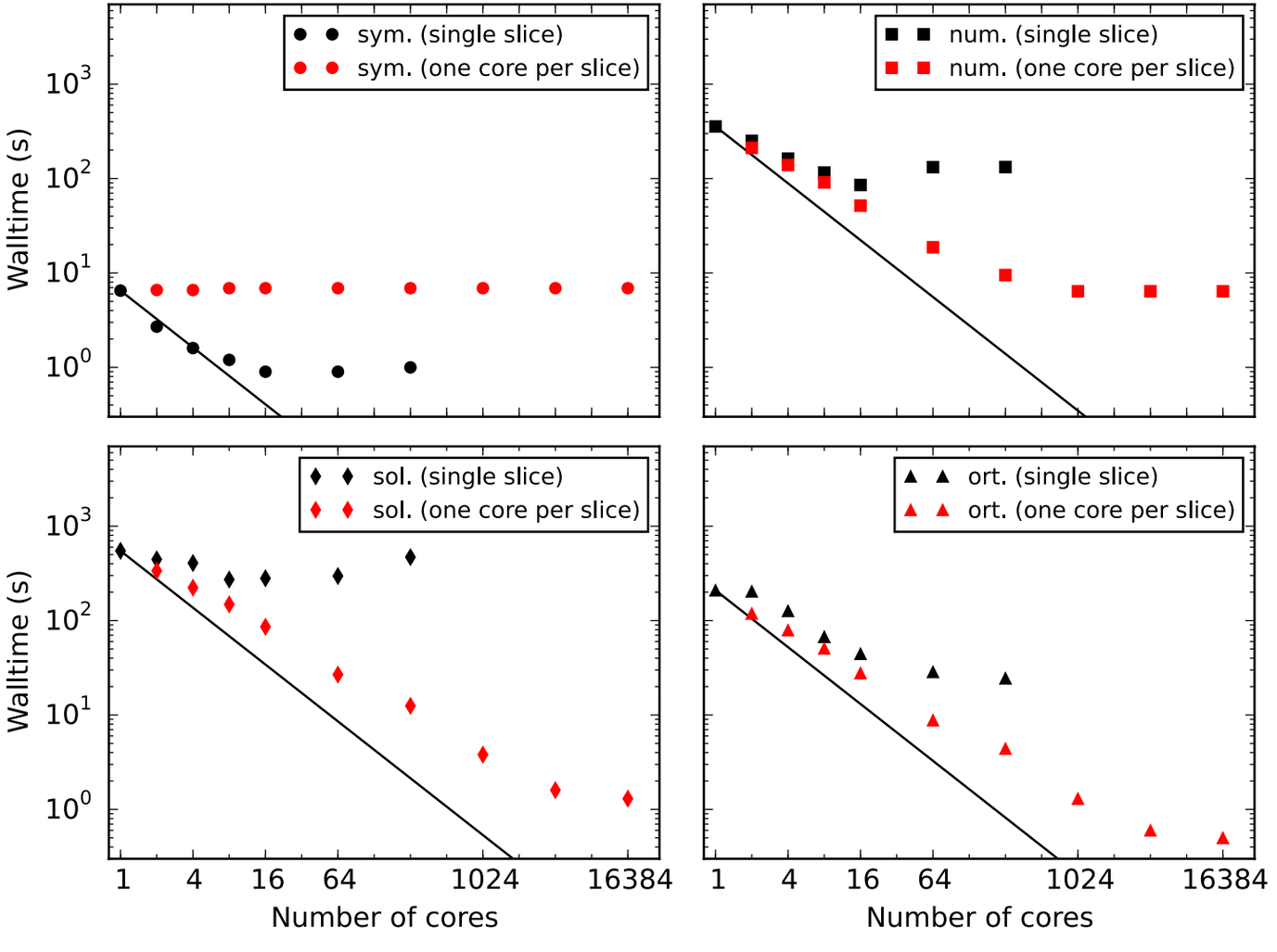
- Only METIS and Scotch has parallel implementations: ParMETIS and PT-Scotch.

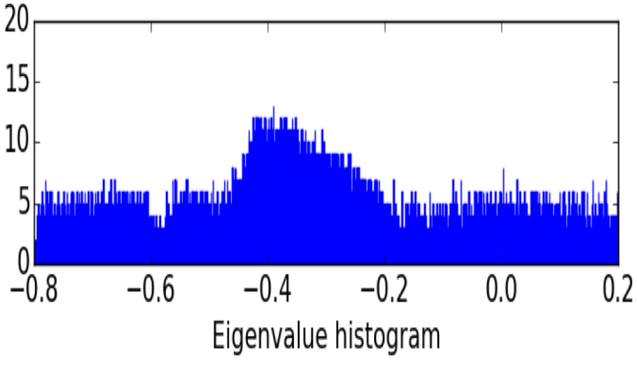
Parmetis vs PT-Scotch

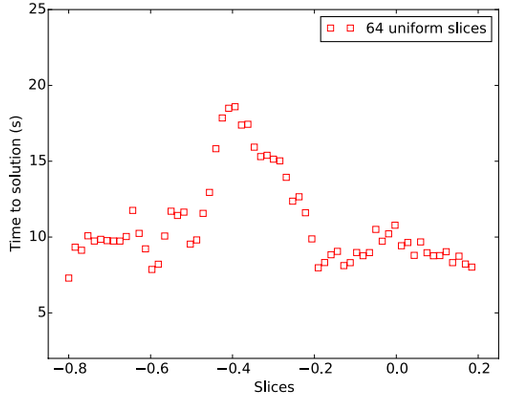


SIPs profile – CNT8000

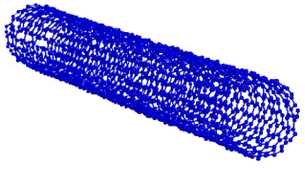


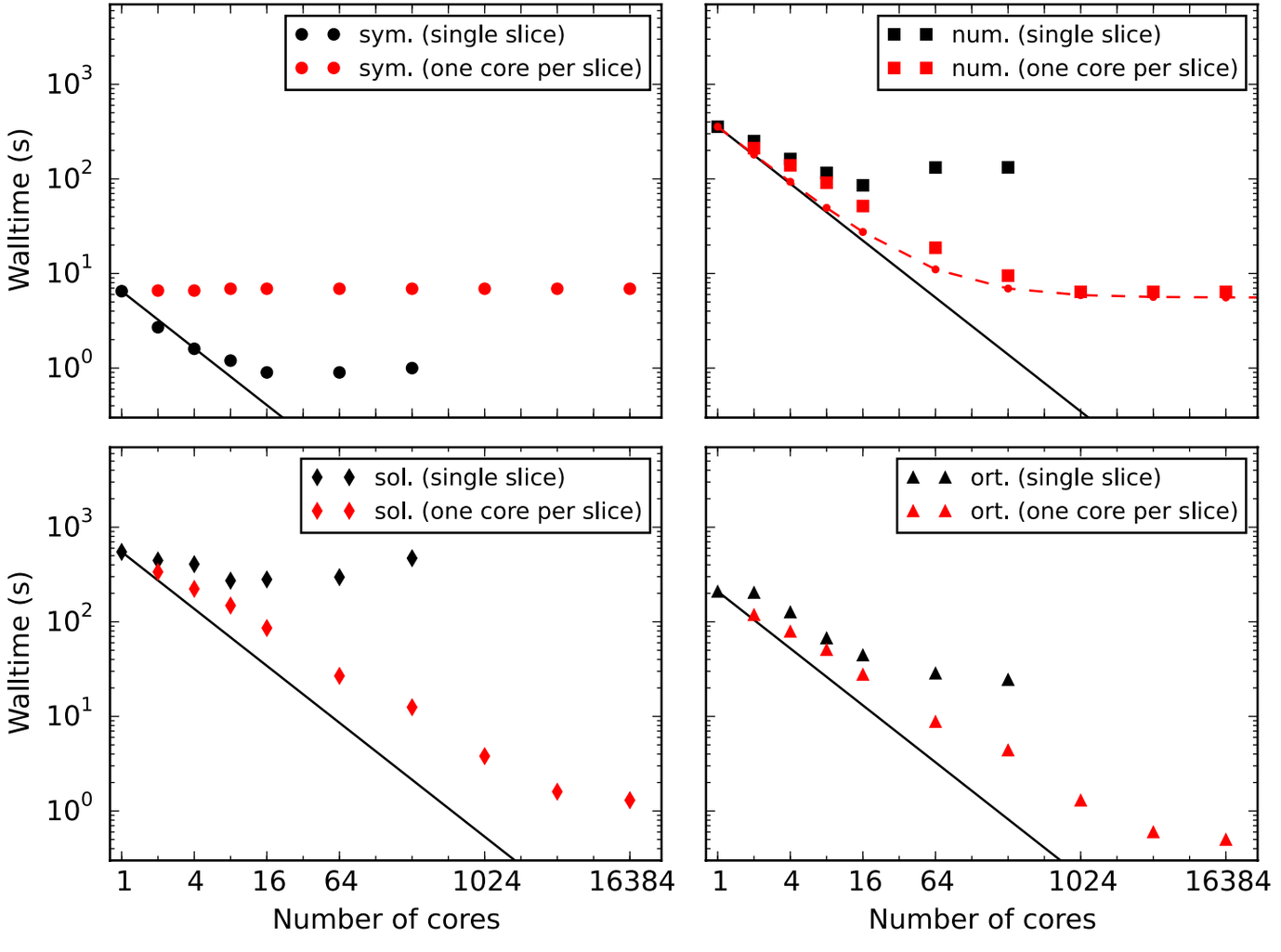






SIPs profile – CNT8000





Amdahl's law:

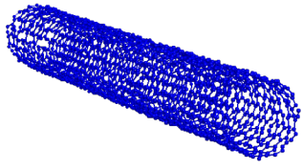
$$t_n = t_1\left(f + \frac{1-f}{n}\right)$$

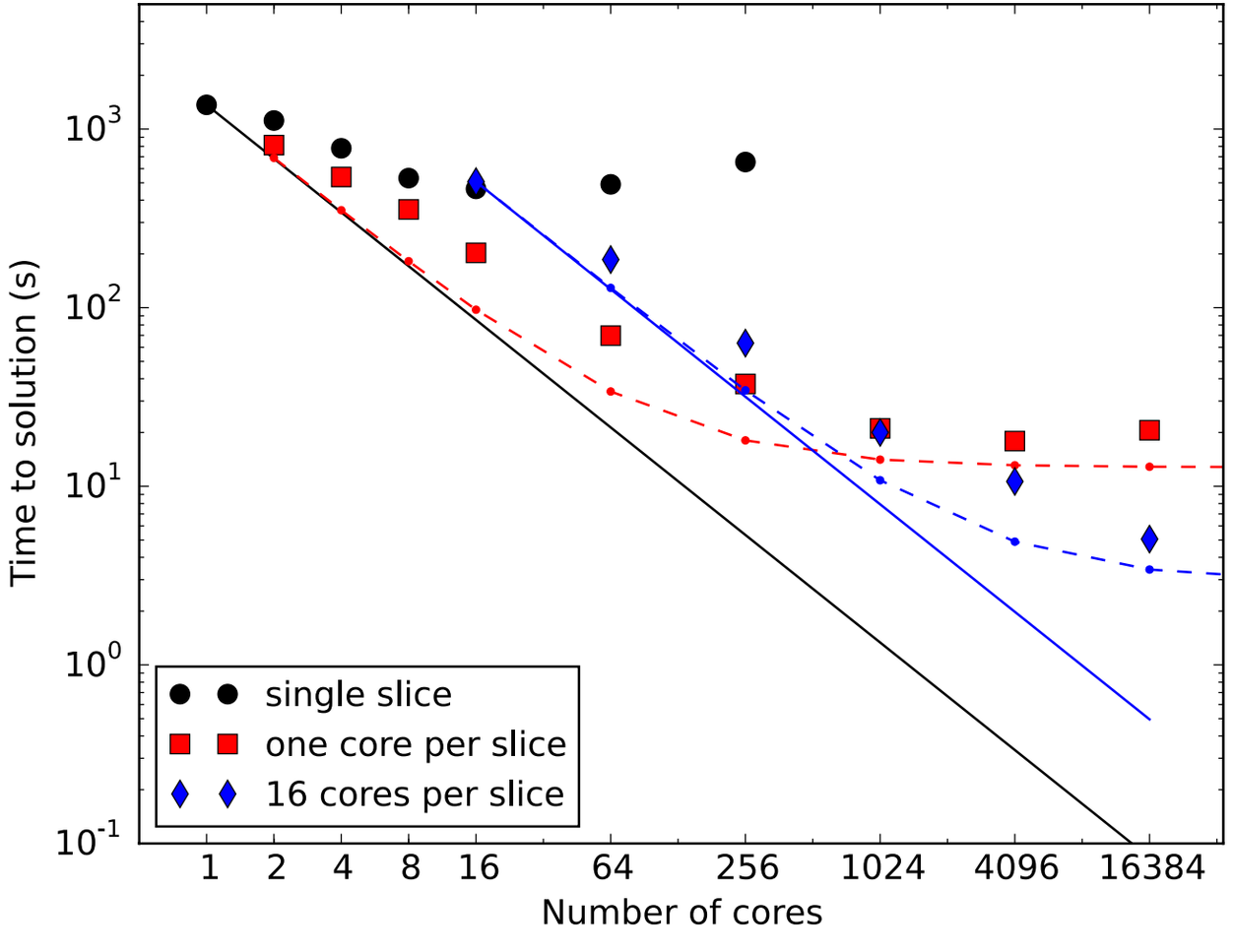
100 sec

90% parallelized

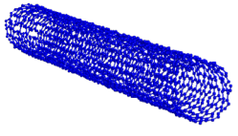
Max speedup x10

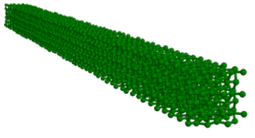
CNT8000

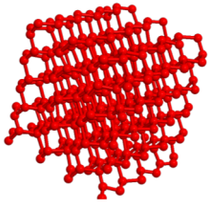


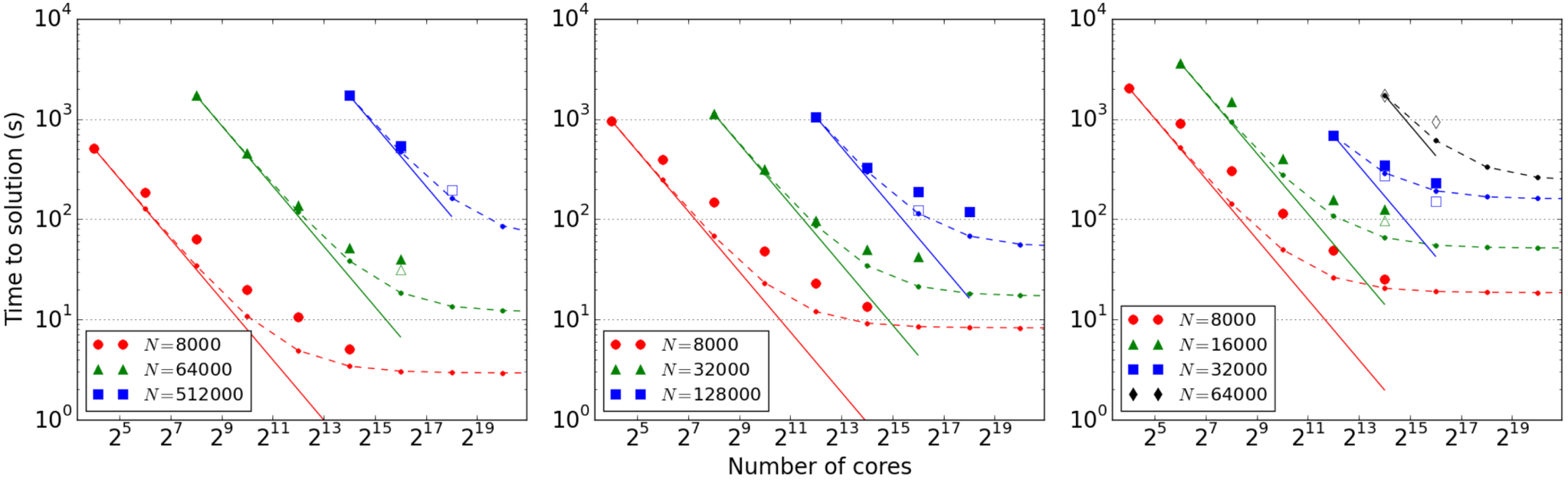

Time to solution (s)
10^3
10^2
10^1
10^0
10^{-1}
single slice
one core per slice
16 cores per slice
1
2
4
8
16
64
256
1024
4096
16384
Number of cores

Strong scaling



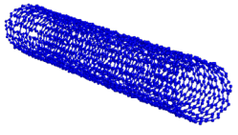


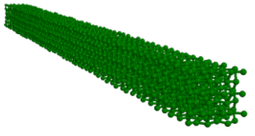


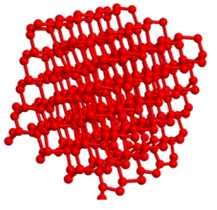


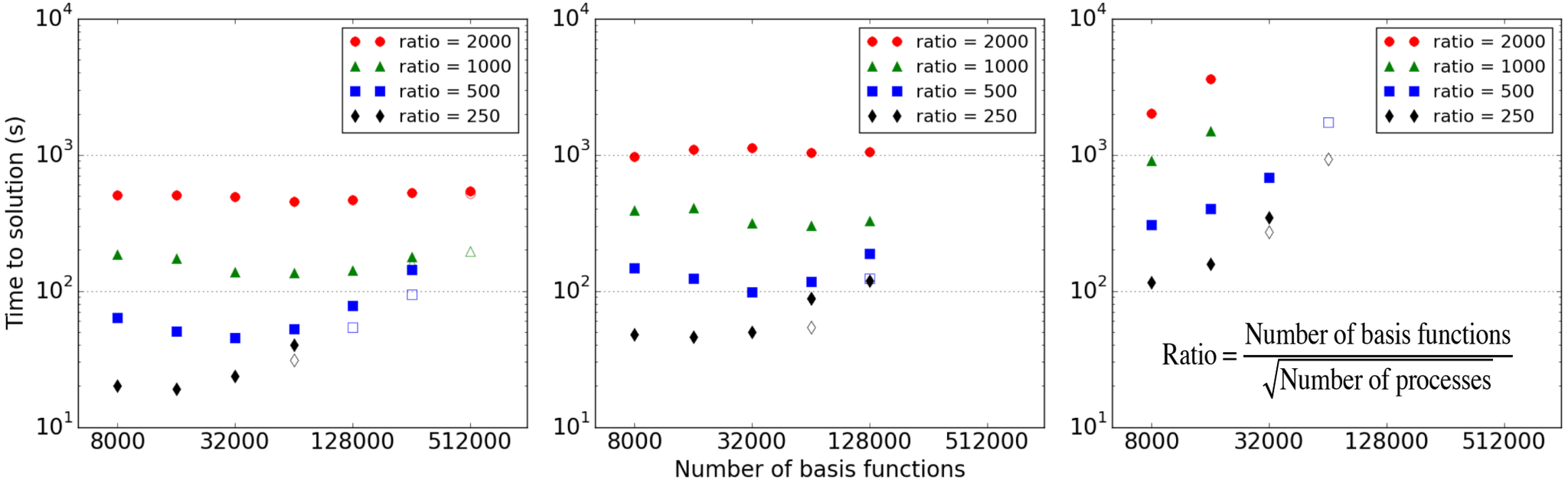
16 cores per slice up to 250,000 cores

Weak scaling





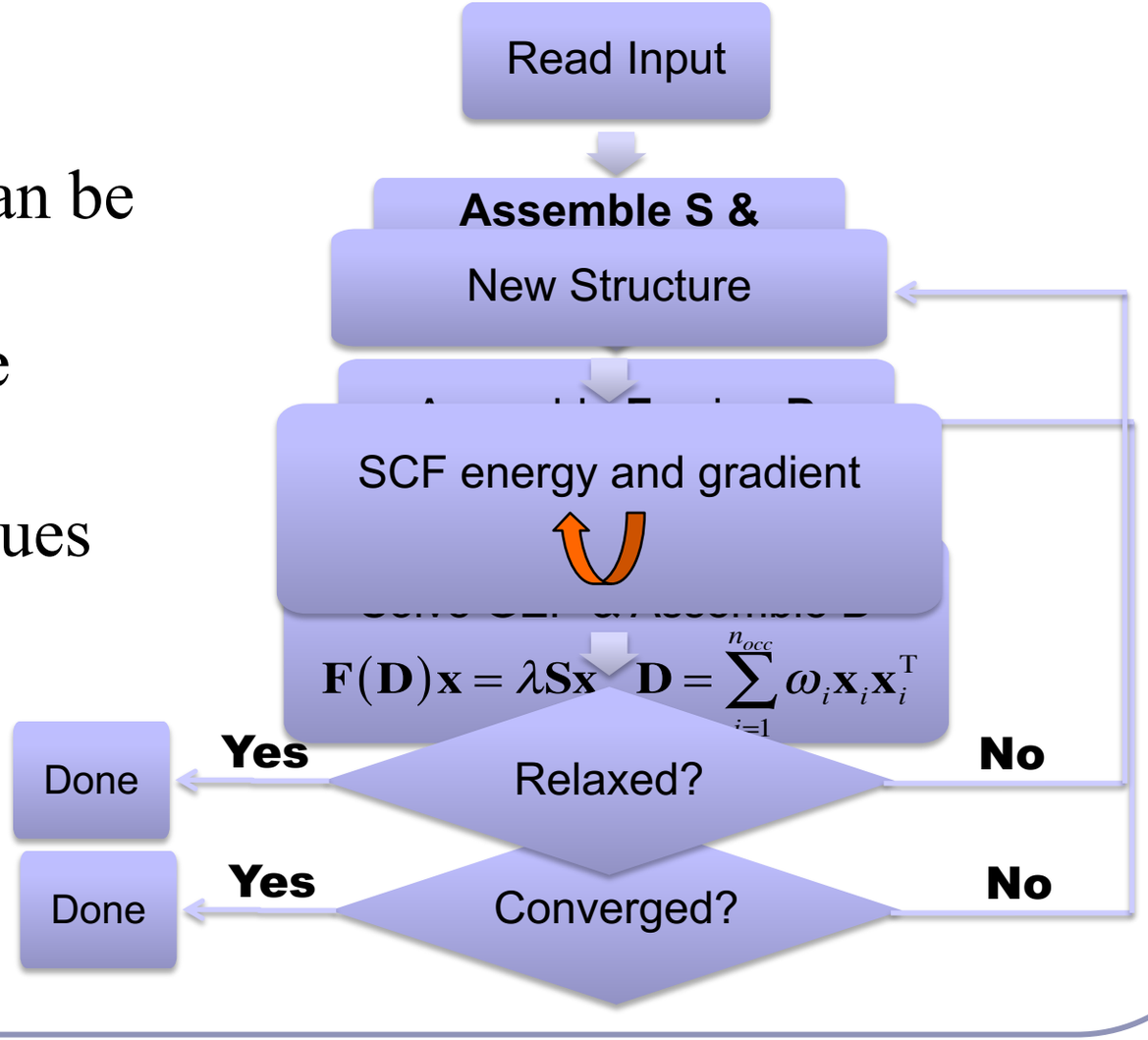




Self-Consistent Field Method

Two distinct advantages for SCF:

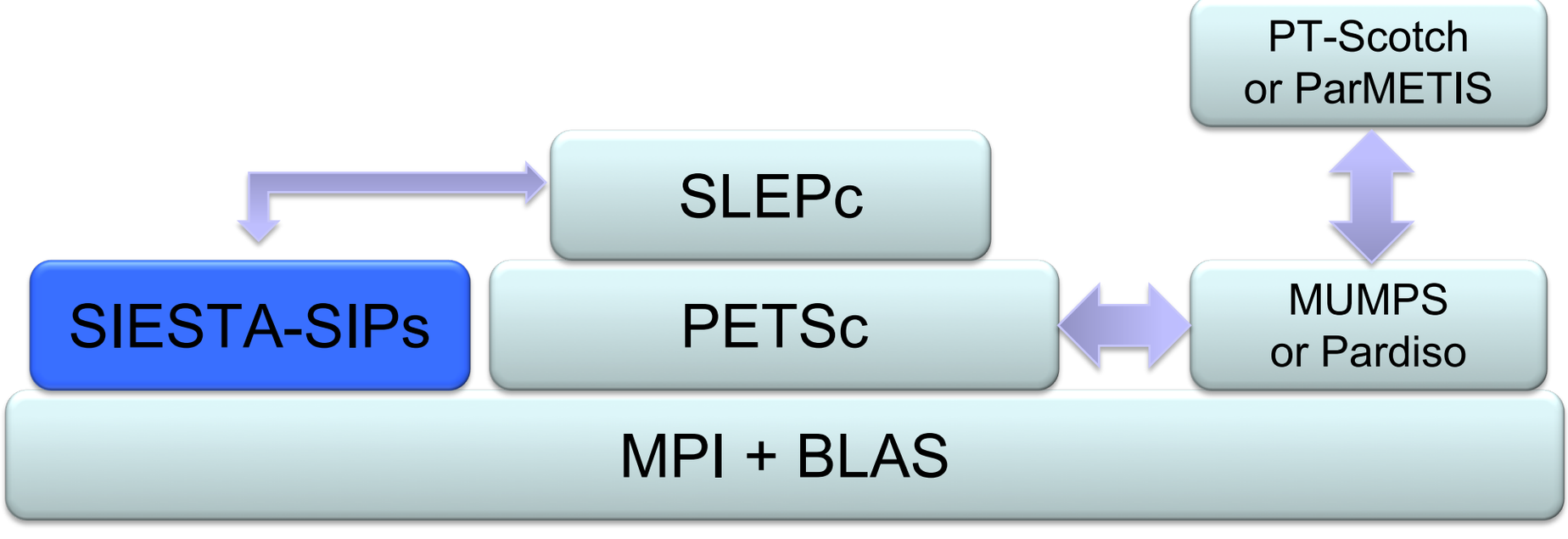
- Symbolic factorization can be avoided since non-zero structure does not change
- Load-balance can be improved as the eigenvalues converge



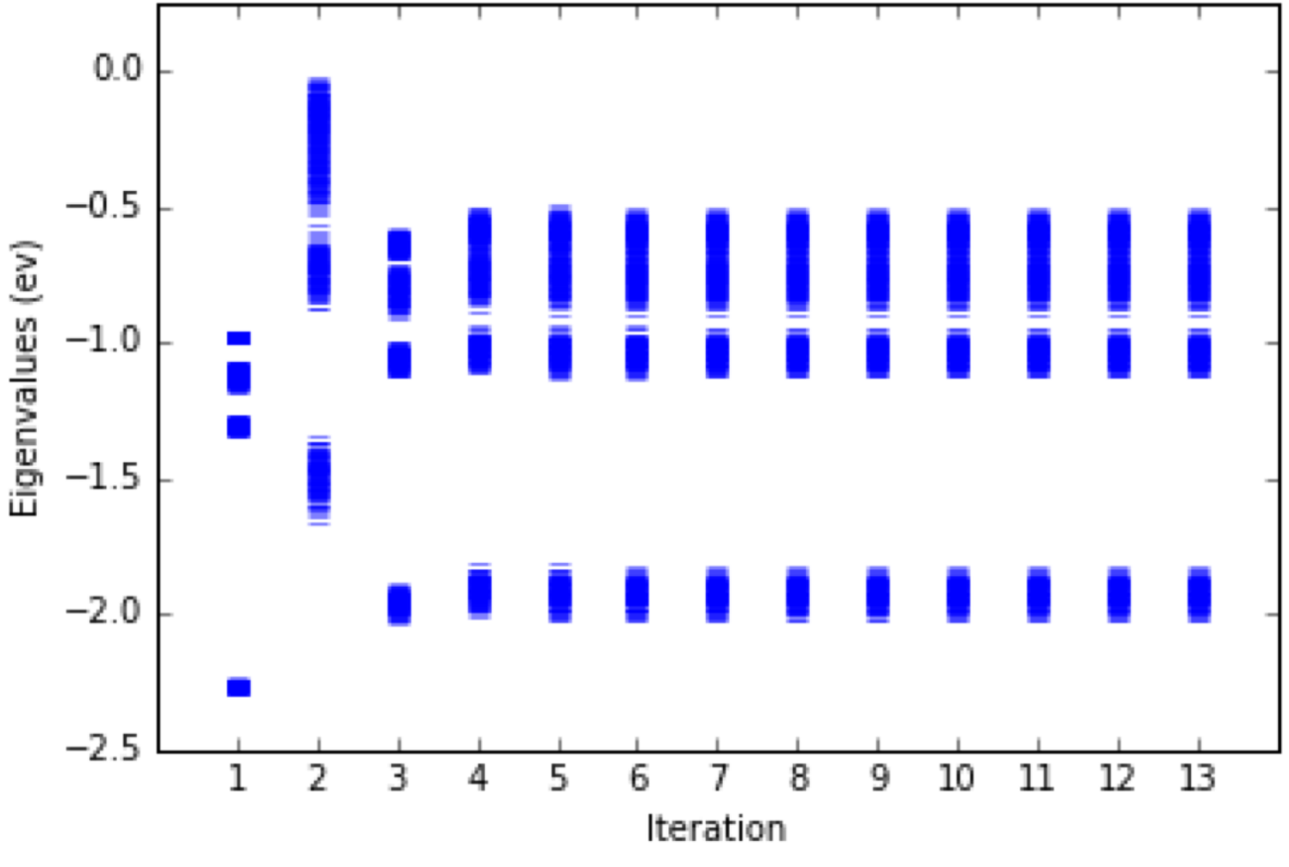
SIESTA Integration

“"SIESTA is both a method and its computer program implementation, to perform efficient electronic structure calculations and ab initio molecular dynamics simulations of molecules and solids.”

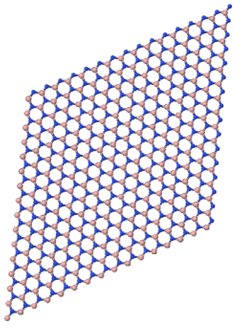
Localized numerical basis sets, sparse Hamiltonian, modular code.



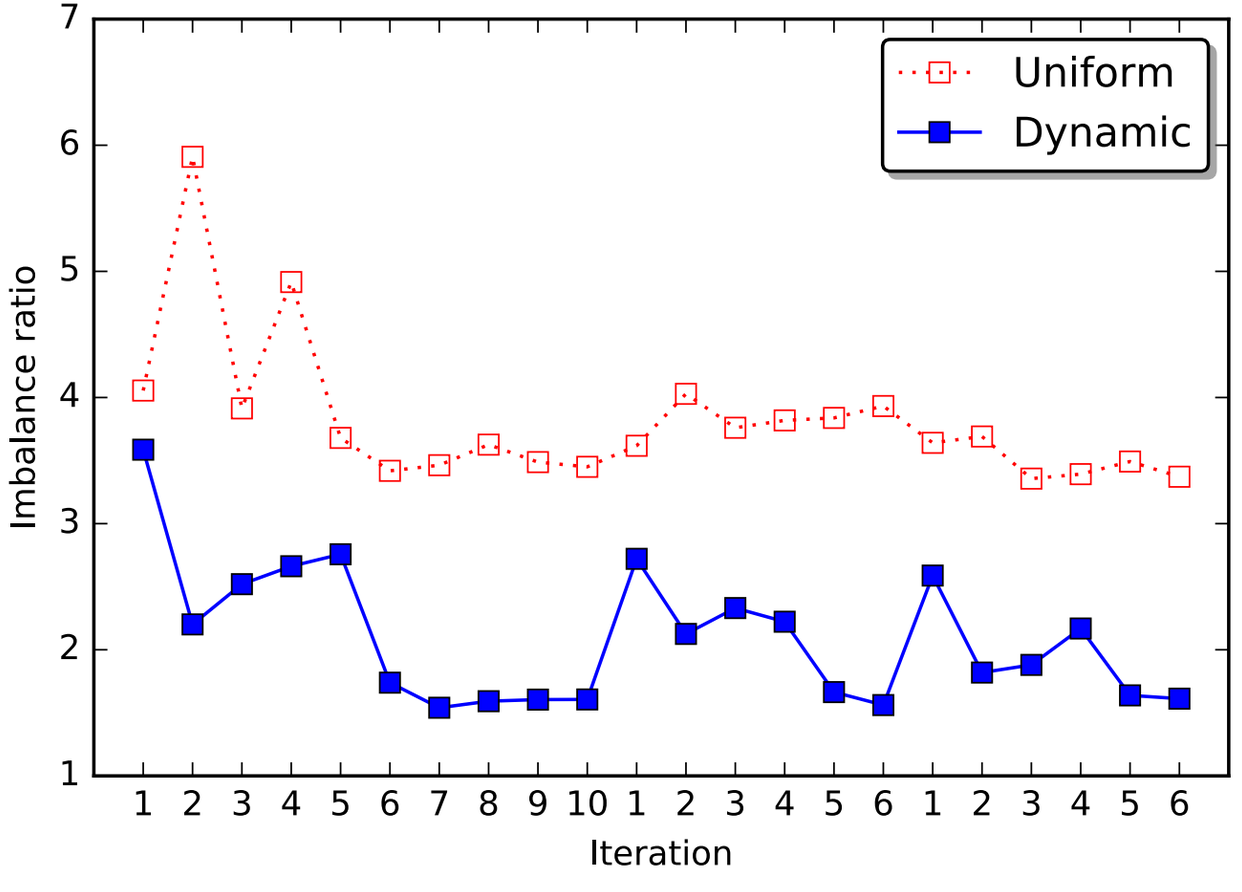
Load balance and eigenvalue distributions



Load Balance



SIESTA DFT CG relaxation for BN monolayer, converged in 3MD steps.

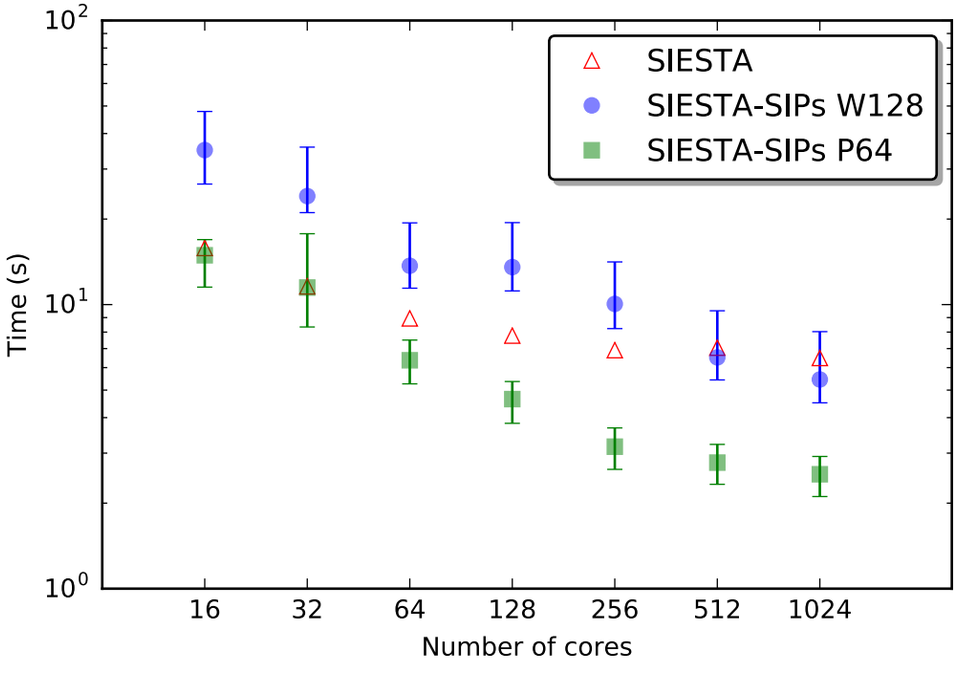


LDA with DZP
512 atoms,
6656 bfs,
nnz 9.2 %

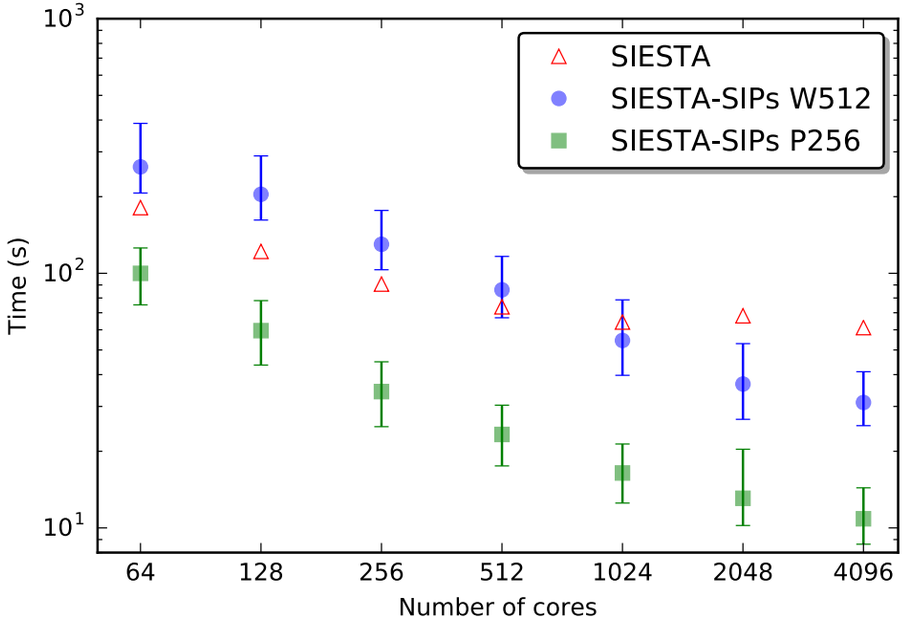
$$\text{Ratio} = \frac{t_{\max}}{t_{\text{mean}}}$$

Benchmark calculations

SIESTA DFT energy and gradient, converged in 13 SCF iterations.
LDA for polyethylene, GGA for water clusters with DZP basis sets



384 atoms, 2944 basis functions
nnz 7%



1536 atoms, 11776 basis functions
nnz 1.8%

Conclusions and outlook for SIPs

- Analyzed the performance and modeled the scaling behavior.
- We demonstrated the strong scaling up to 250k and 500k basis functions
- A new implementation is available through SLEPc and also available in ELSI.
- Performance is better if you don't have a gap unlike other fast solvers and improves as SCF converges.
- Γ-point only, performance decreases with more multiplicities.
- At the strong scaling limit (1 factorization per slice, up to 40 eigenpairs) should be compatible with PEXSI.
- Next steps:
  - Report/fix bug that prevents us use MKL
  - Estimate eigenvalue distribution with cheaper methods.
  - MKL-Pardiso integration, requires PETSc/MKL
  - Collaboration through ALCC, INCITE, ADSP

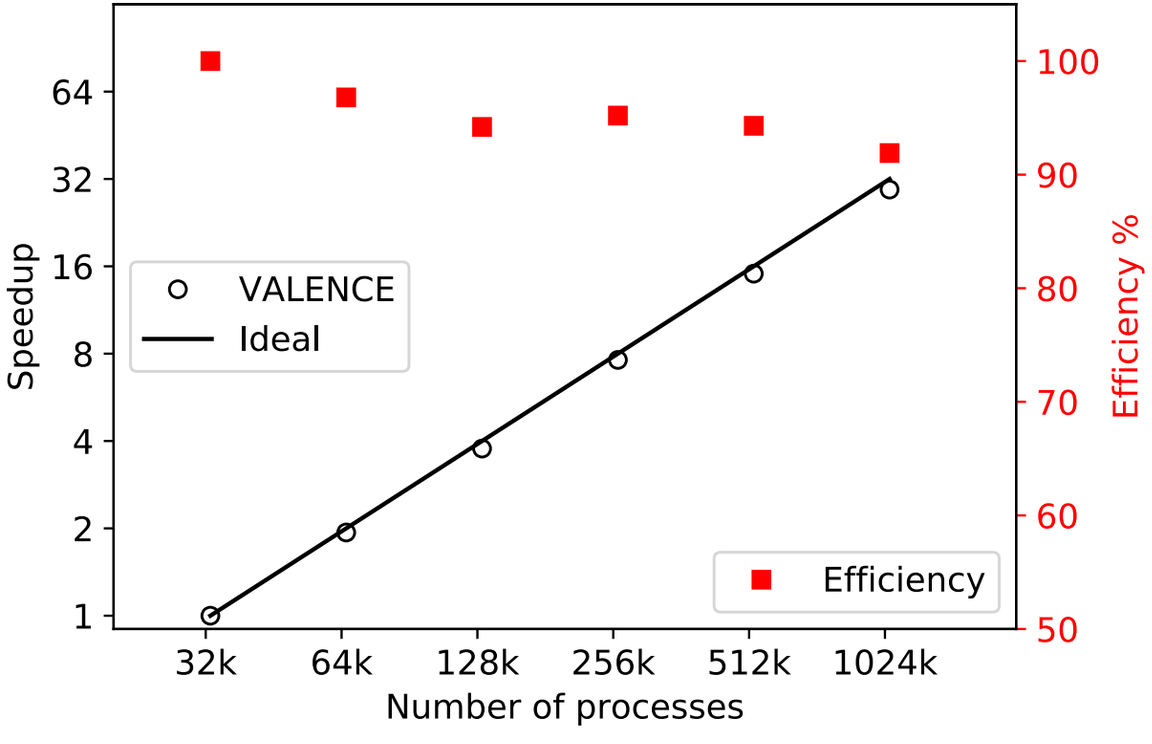
VALENCE

- Based on valence bond theory and overlapping (nonorthogonal) linear combinations of atomic orbitals (OLCAO)
- Soon to be released.

Graham Fletcher (ANL)

Colleen Bertoni (ANL)

Michael D'Mello (Intel/ANL)



Acknowledgment

Hong Zhang, Peter Zapol, Al Wagner, Âlvaro Vâzquez-Mayagoitia (ANL)

Jeff Hammond (Intel)

David Dixon (U of Alabama)

Jose Roman, Carmen Campos (Universitat Politècnica de València)

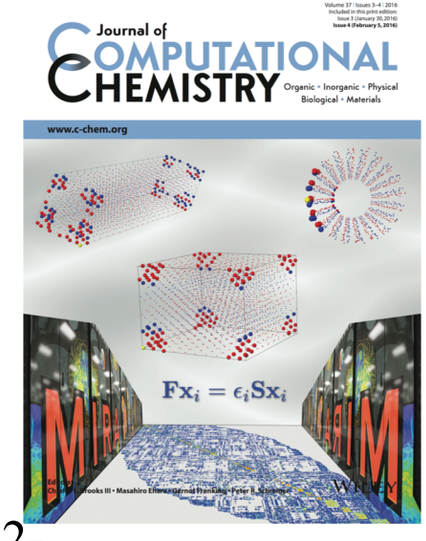
Fabiano Corsetti (Imperial College London)

Victor Yu, Volker Blum, (Duke U)

Thanks to PETSc, SLEPc, Elemental, MUMPS, and PT-Scotch developers.

Work supported by US DOE, Office of Science under Contract No. DE-AC02-06CH11357.

Computations are done on Blues, Vesta, Mira, Theta at ANL, Cori at NERSC.

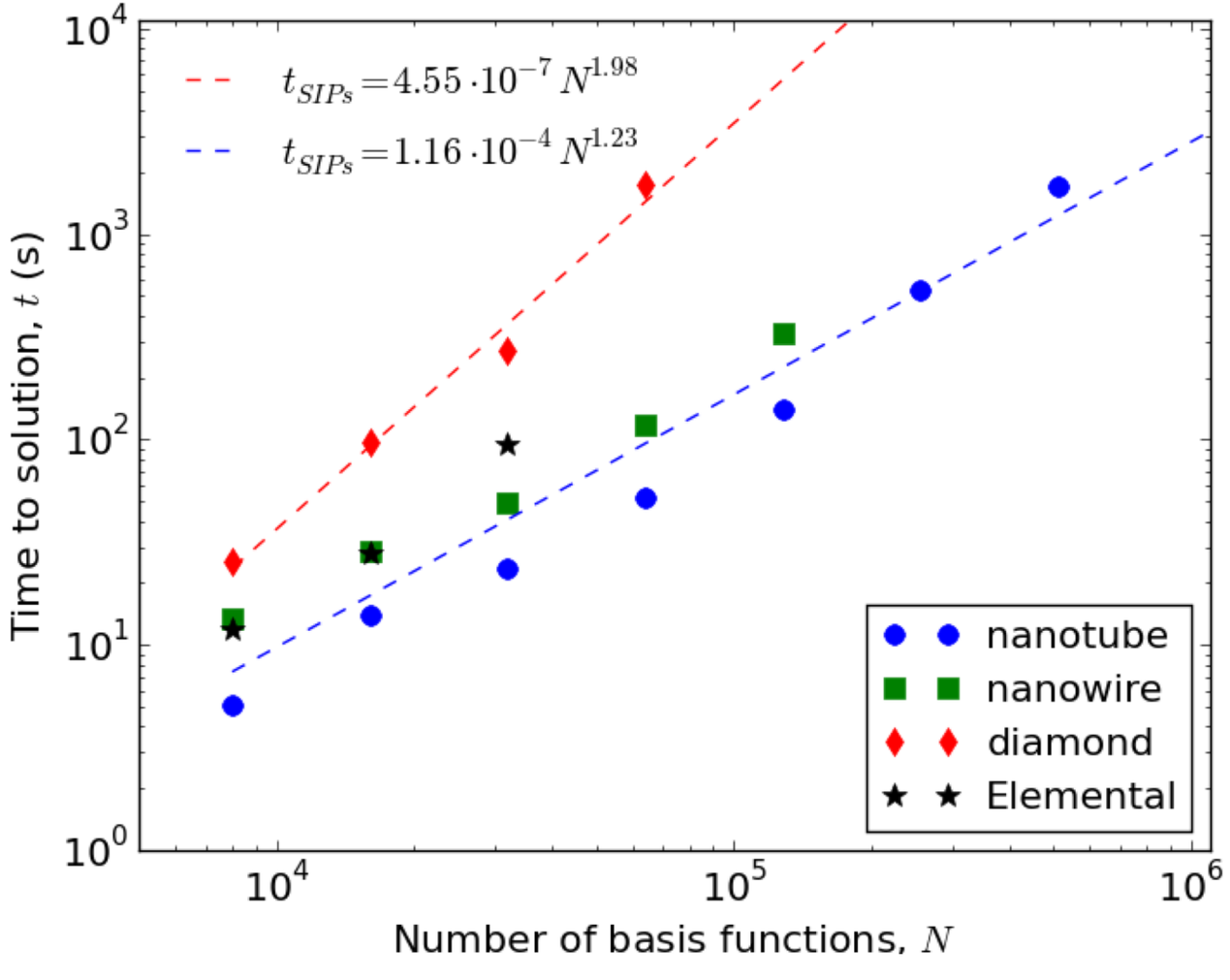


1. Zhang, H., Smith, B., Sternberg, M. & Zapol, P. SIPs. *ACM Trans. Math. Softw*. 33, 9 (2007).
2. Campos, C. & Román, *J. E. Numer. Algorithms*, 60, 279 (2012).
3. Keçeli M., Zhang, H., Zapol, P., Dixon A.D., & Wagner A. F., *J. Comput. Chem*. 37, 448 (2016).
4. Keçeli M., Corsetti F., Campos C., Roman J., Vâzquez-Mayagoitia A, Zhang, H., Zapol, P., & Wagner A, F. *J. Comput. Chem*. (2018).

SIPs vs Elemental

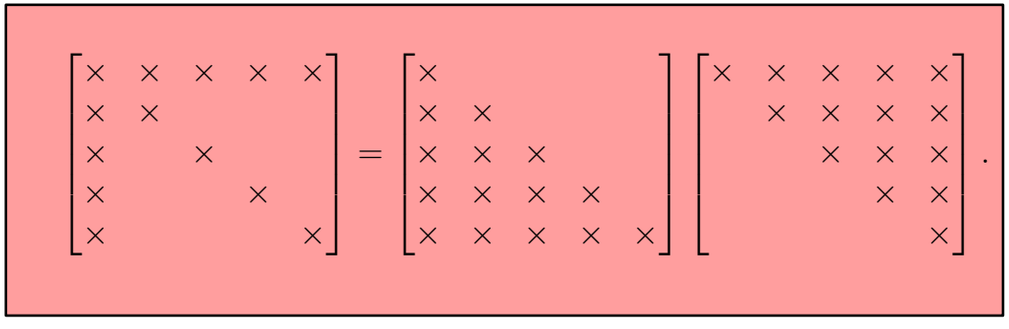
Fixed resources

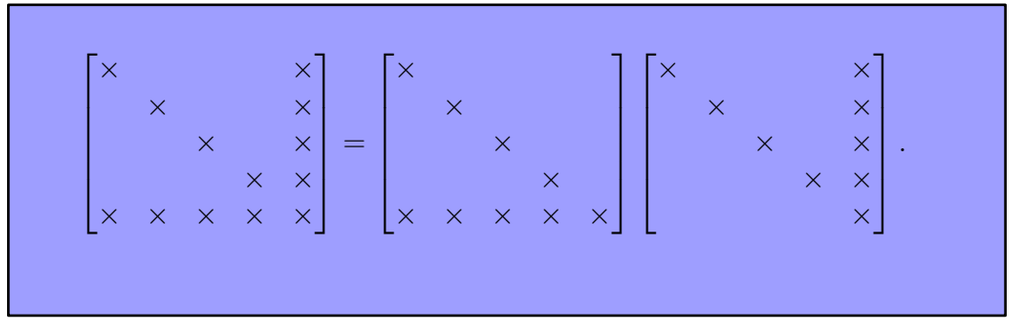
16,384 cores



Fill-ins due to factorizations

- Ordering of the original matrix is crucial to minimize fill-ins

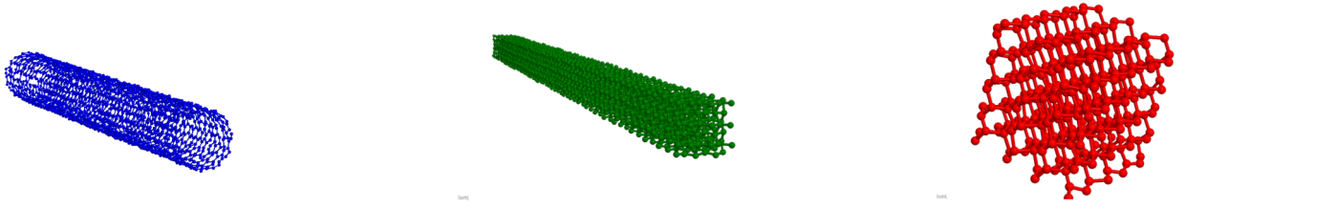




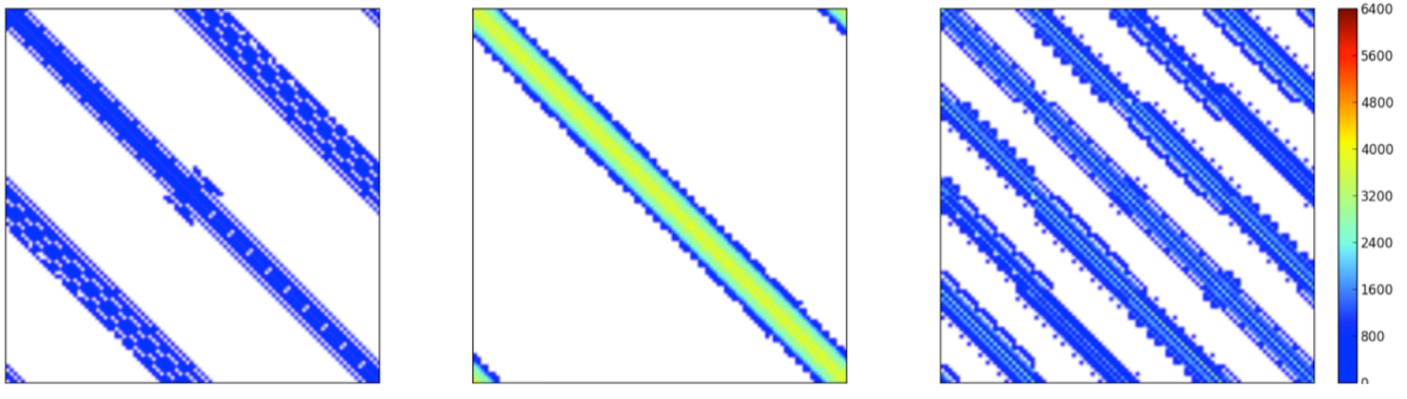
- How can we find the best ordering to minimize fill-ins?
  - Computing the minimum fill-in is NP-complete.
    - No deterministic algorithm exists to find it in polynomial time.
    - There are a number of heuristic algorithms based on graph theory for approximate solutions.

*M. Yannakakis. SIAM J. Algebraic Discrete Methods, 2:77-79,1981

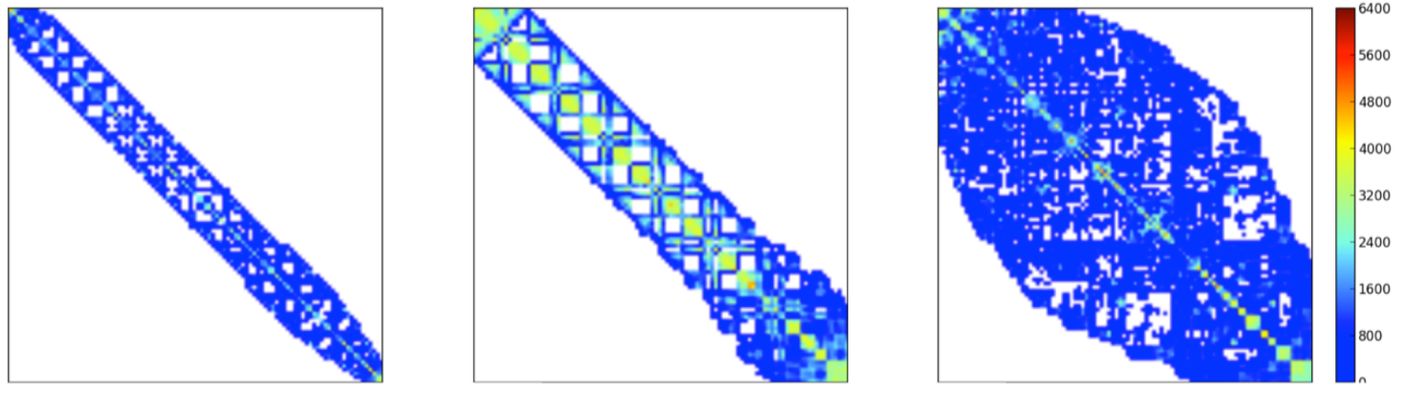
Reordering matrices



Original



Reordered



SIESTA-SIPs applications