Towards Large Scale Quantum Chemistry Calculations on the K and Fugaku Supercomputer

William Dawson
Takahito Nakajima
RIKEN Center for Computational Science

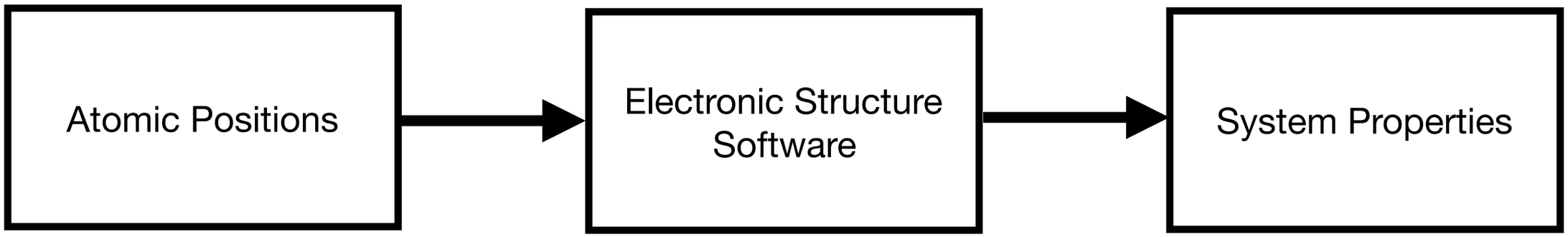
ADAC8 Workshop, University of Tokyo, Kashiwa, Japan.

Computational Molecular Science Research Team

- Also in collaboration with Luigi Genovese (French Alternative Energies and Atomic Energy Commission).

Theme: Chemical Properties from First Principles

- The goal of our research is to develop theoretical methods, numerical algorithms, and software to study the properties of molecules and materials from the basic laws of the universe (in particular: quantum mechanics).

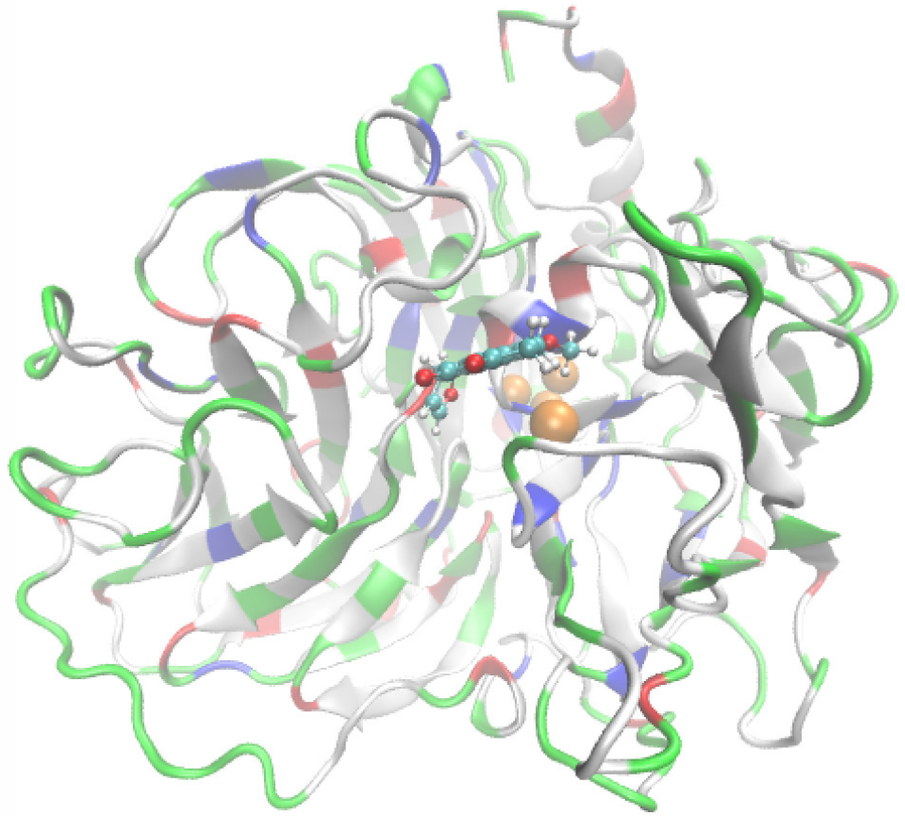


- Electronic structure methods allow for a quantum mechanical treatment of systems, giving scientists tools to go beyond the technical limits of experiment in understanding the natural world and designing new materials.
- A broad field with many different methods and algorithms.

Example Application: Bioremediation

- Bioremediation: the use of naturally occurring organisms to break down environmental pollutants.
- Example: using enzymes to degrade toxins, which can contaminate food supplies.

- Requirement 1: computational methods and software that can perform accurate calculations on such large systems (10,000 atoms).
- Requirement 2: methodologies that can extract scientific insight from these calculations.



Zaccaria, Marco, WD, Viviana Cristiglio, Massimo Reverberi, Laura E. Ratcliff, Takahito Nakajima, Luigi Genovese, and Babak Momeni. "Designing a bioremediator: mechanistic models guide cellular and molecular specialization." Current opinion in biotechnology 62 (2020): 98-105.

Requirement 1: Methods for Large Systems

- One of the major computational bottlenecks for quantum chemistry software is computing the density matrix D from the hamiltonian H.
- This is typically done by solving an eigenvalue problem.
  - $HV = \lambda V$ (where V is a n_{basis} x $n_{electrons}$ matrix).
  - Then compute the density matrix as $D = VV^T$.
    - Computational cost scales with the third power of the system size.
- Instead, we can compute the density matrix directly using the fermi function:
  - $D = 1/(e^{\beta(H-\mu)} + I)$.
- This function here is a matrix function. Is it possible to use this framework to reduce the computational cost?

Introduction to Matrix Functions

- Simple definition: We are all familiar with functions of a single variable f(x). In the matrix function case, just replace the variable x with a matrix A.

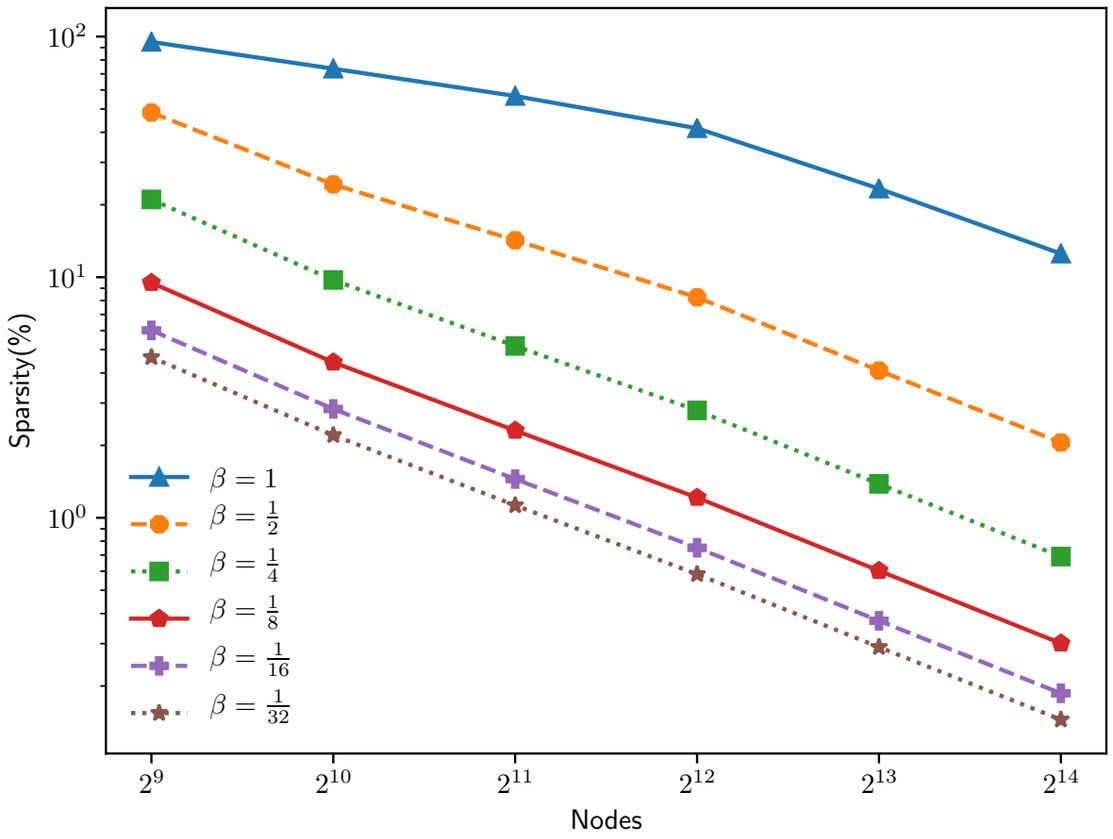
Standard Function	**Matrix Function**	**Interpretation**
$f(x) = x^2$	$f(A) = A^2$	Matrix Product
$f(x) = 1/x$	$f(A) = A^{-1}$	Matrix Inverse
$f(x) = e^x$	$f(A) = e^A$	Matrix Differential Equation
$f(x) = \text{sign}(x)$	$f(A) = \text{sign}(A)$	Projection on to Subspace

- Matrix functions have many different applications including solving differential equations and the study of complex networks.

Higham, Nicholas J. *Functions of matrices: theory and computation*. Vol. 104. Siam, 2008.

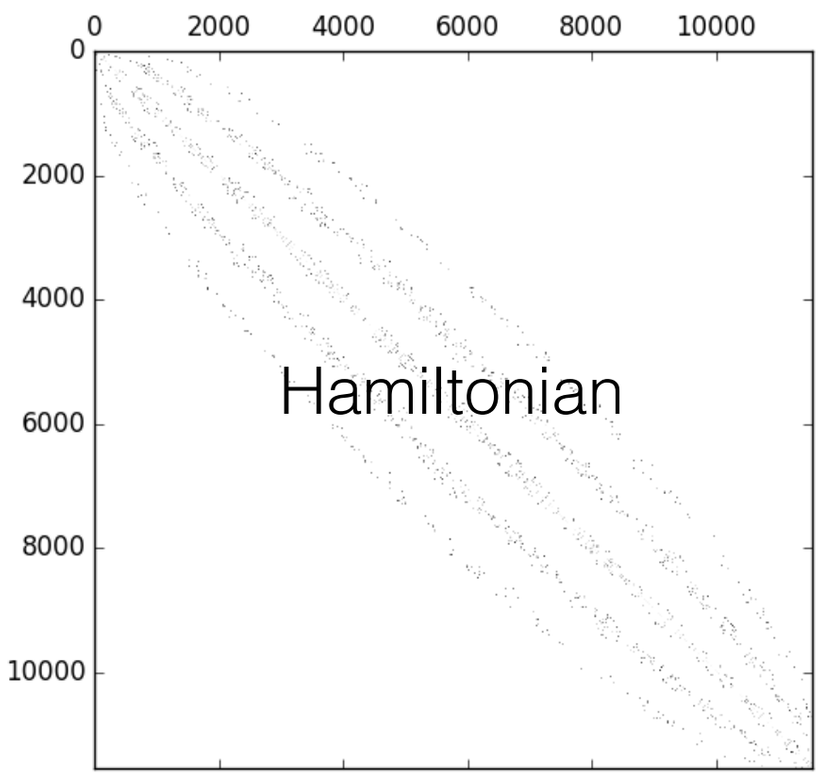
Motivating Matrices

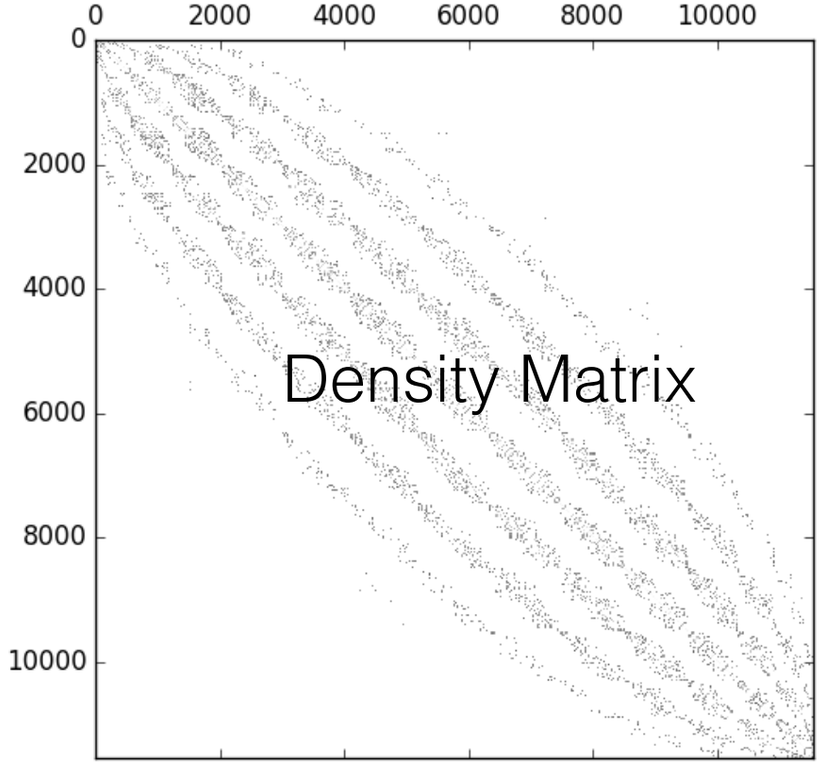
- In many domains, the problem of interest can be represented using a sparse, hermitian matrix.
- Under certain conditions, not only is the matrix A sparse, but also some matrix functions f(A) are sparse.
- Estrada matrix exponential $e^{\beta A}$ contains a scaling factor which might be interpreted as a unit of edge weight.
- For certain values of β, the matrix exponential of small world matrices is also sparse.



Motivating Matrices - Chemistry

- For insulating systems (and metals at high temperature), it is known that the density matrix is sparse when represented in a localized basis.
- Example: the Hamiltonian and Density Matrix of 1920 water molecules computed using BigDFT.



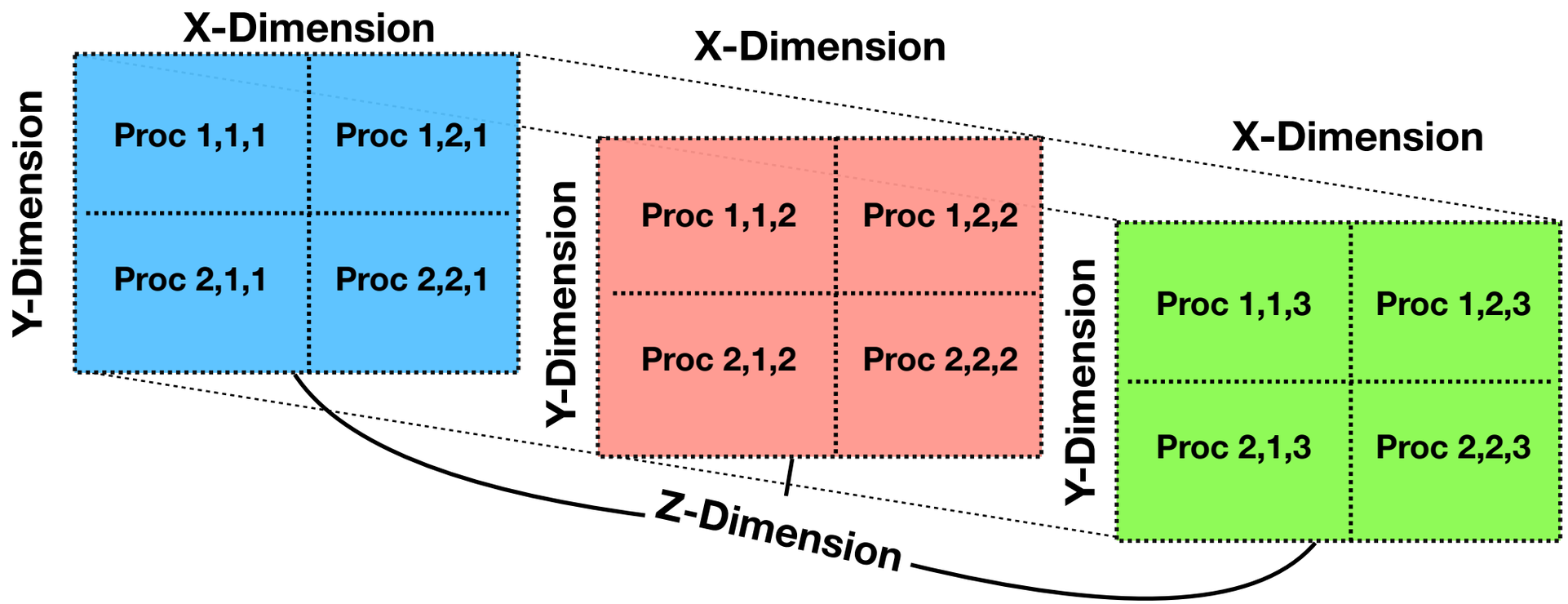


Sparsity Aware Matrix Function Calculation

- From the list of methods for computing matrix functions, we will select calculations based on matrix polynomials.
  - e.g. Chebyshev polynomials: $f(A) \cong \sum c_i T_i(x)$.
  - $T_0(A) = I$ $\quad T_2(A) = 2A^2 - I$ $\quad T_4(A) = 8A^4 - 8A^2 + I$
  - $T_1(A) = A$ $\quad T_3(A) = 4A^3 - 3A$
- Computing a matrix polynomial requires only two core routines: matrix addition, matrix multiplication.
  - Easy to parallelize.
  - Many functions can be tuned through just two routines.
- In the case of sparse matrices, we replace these kernels with sparse matrix addition, and sparse matrix multiplication.

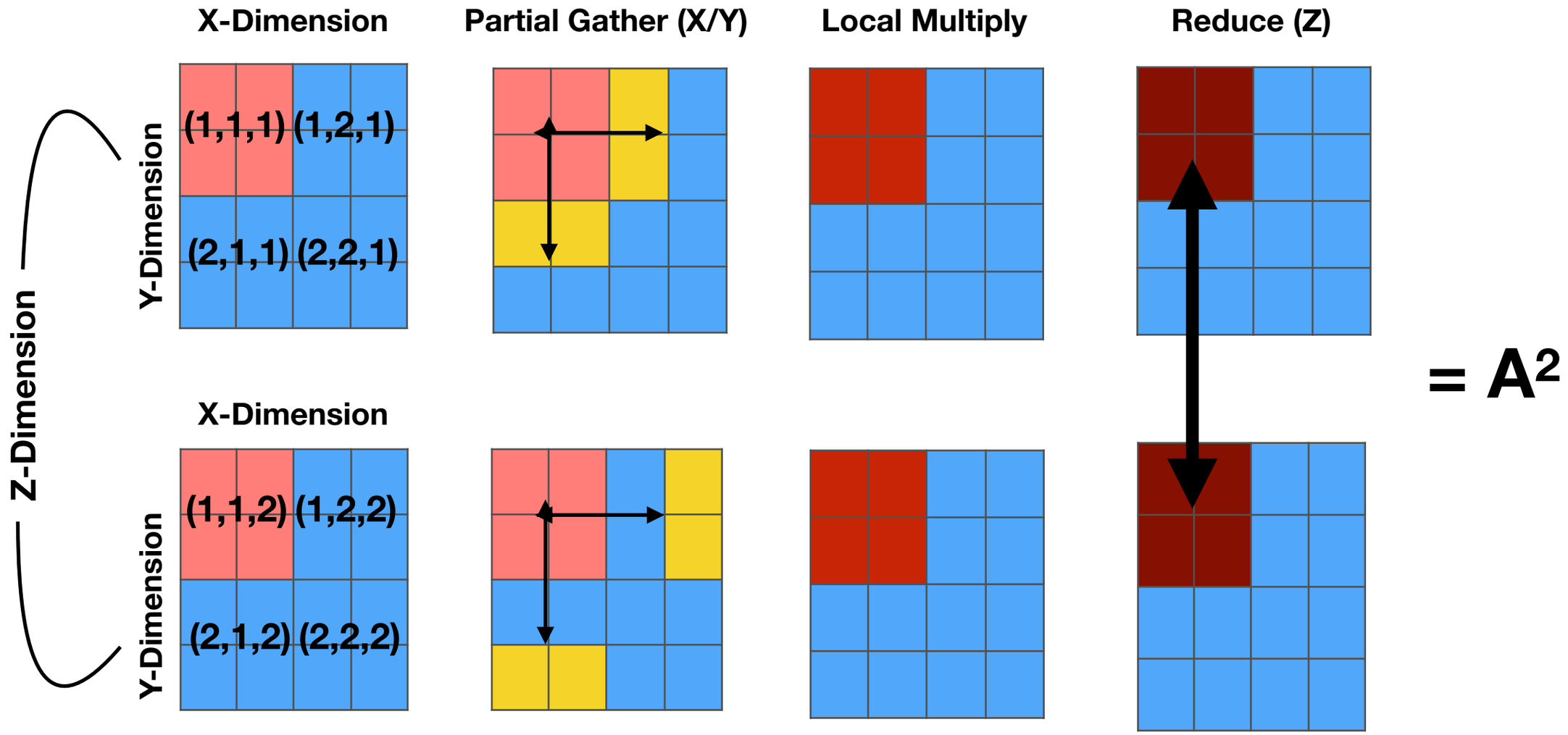
Parallel Implementation: Topological Aware

- Approach: a three dimensional data distribution maps well to the high level 6D Tofu Network topology of K/Fugaku.
- Map a 2D matrix onto a 3D topology, and exploit the extra 3 dimensions for efficient collective communication.
- For each dimension, we are free to choose whether to partition or replicate data structures.



Parallel Implementation: Matrix Multiplication

- Core routine of computing matrix powers.
- Overdecomposition of the matrix enables strong scaling performance.
- Gather+Reduce implemented as collective communication.

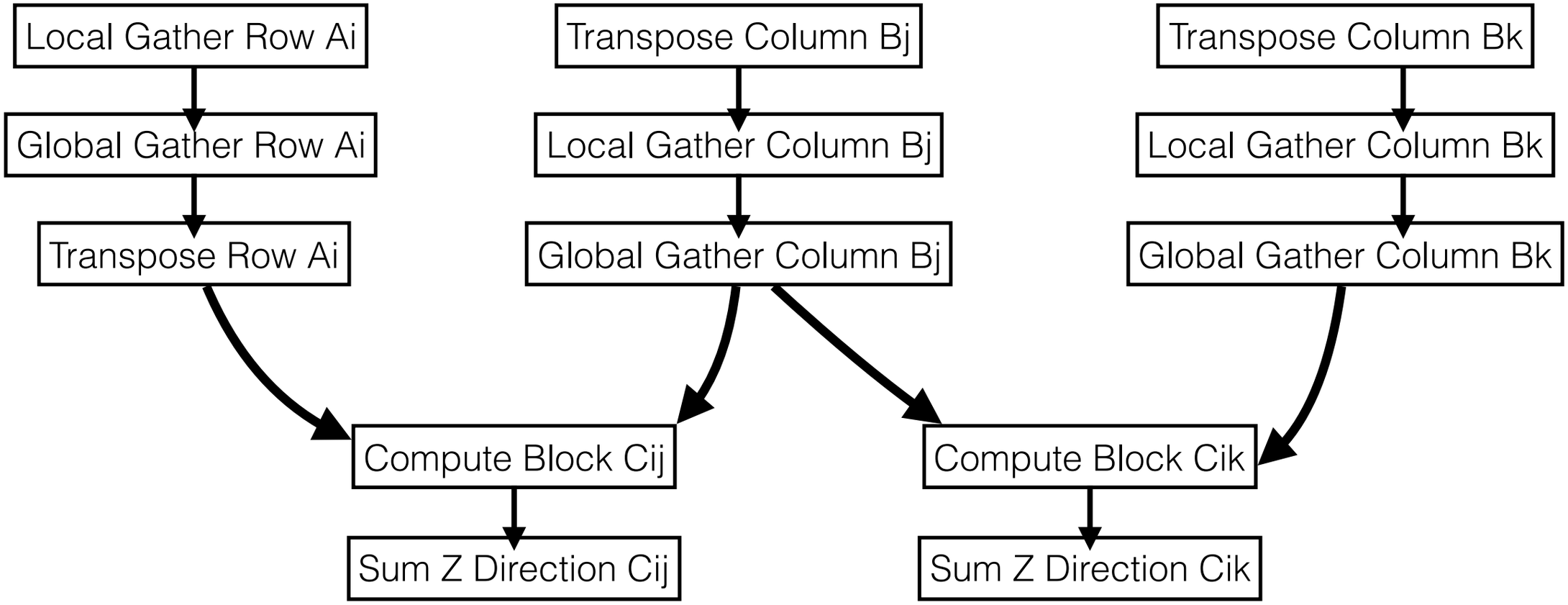


[1] Solomonik, Edgar, and James Demmel. "Communication-optimal parallel 2.5 D matrix multiplication and LU factorization algorithms." In *European Conference on Parallel Processing*, pp. 90-109. Springer, Berlin, Heidelberg, 2011.

[2] Dawson, William, and Takahito Nakajima. "Massively parallel sparse matrix function calculations with NTPoly." *Computer Physics Communications* 225 (2018): 154-165.

Parallel Implementation: Thread Parallel

- OpenMP loop parallelism doesn't work well with overlapping communication. Instead we use OpenMP task framework.
- Creating a task manager, and dependency graph.
- Block levels tasks help mitigate load imbalance from sparse matrix and communication in different directions.



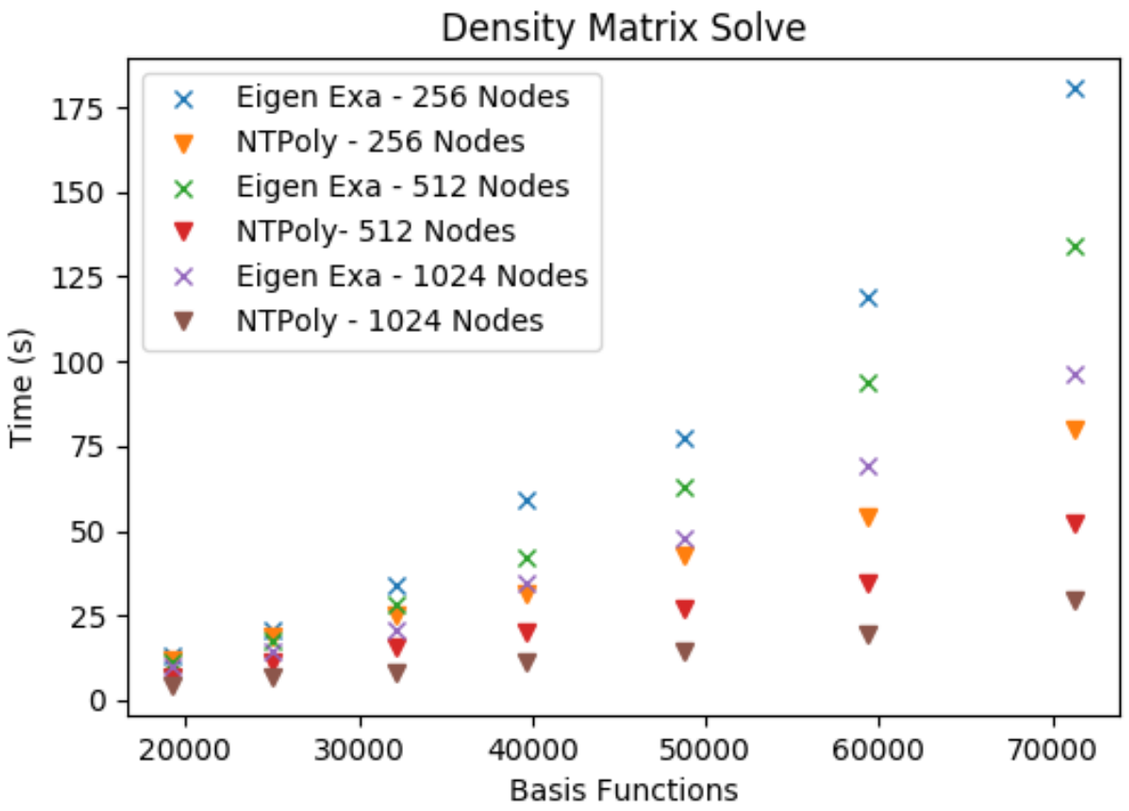
NTPoly Features

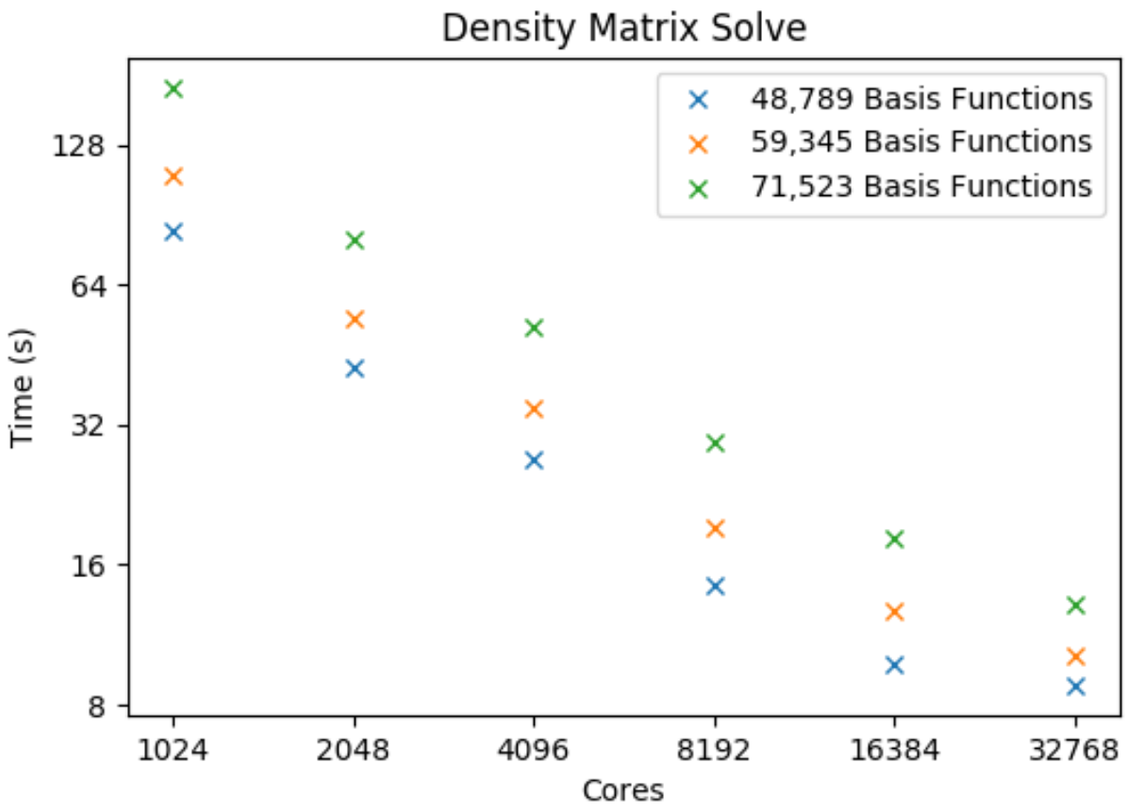
- **General Polynomials**
  - Standard Polynomials
  - Chebyshev Polynomials
  - Hermite Polynomials
- **Transcendental Functions**
  - Trigonometric Functions
  - Exponential and Logarithm
- **Matrix Roots**
  - Square Root and Inverse Square Root
  - Matrix *p*th Root and Inverse *p*th root
  - Matrix Inverse (and Moore-Penrose Inverse)
  - Sign Function/Polar Decomposition
- **Quantum Chemistry**
  - Density Matrix Purification
  - Chemical Potential Calculation
  - Density Matrix Extrapolation
- **Other**
  - Parallel File I/O
  - MIT License
  - Fortran/C/C++/Python Interface
  - Automatic data redistribution
  - Real and Complex Matrices

WD, and Takahito Nakajima. "Massively parallel sparse matrix function calculations with NTPoly." *Computer Physics Communications* 225 (2018): 154-165.

Performance: Quantum Chemistry

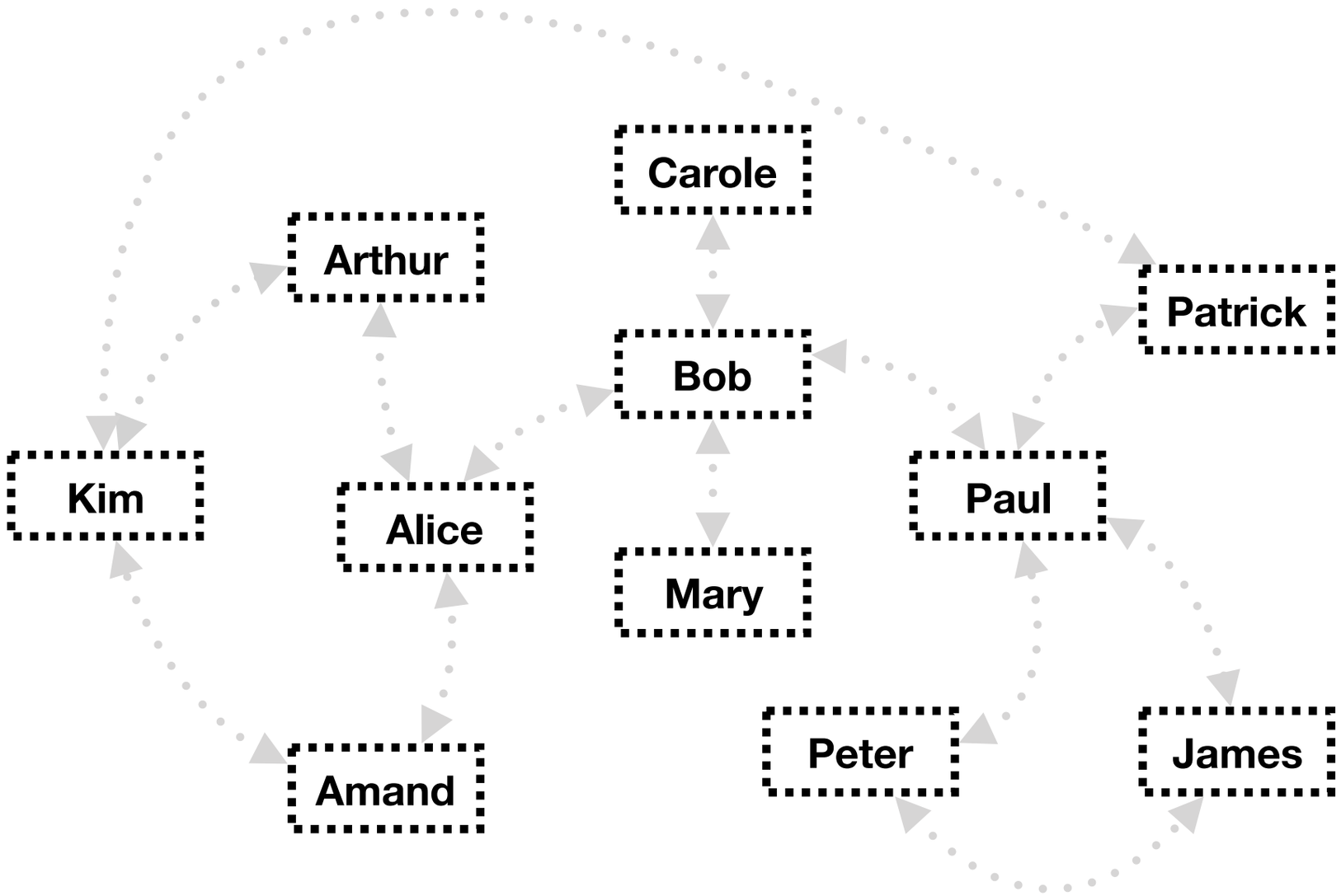
- Standard eigensolvers can make limited use of the sparsity of a matrix, but will be outperform by matrix function based approaches.
- Calculation of water clusters of various sizes, 6-31G basis set, using the TRS2 density matrix method to approximate the fermi function.

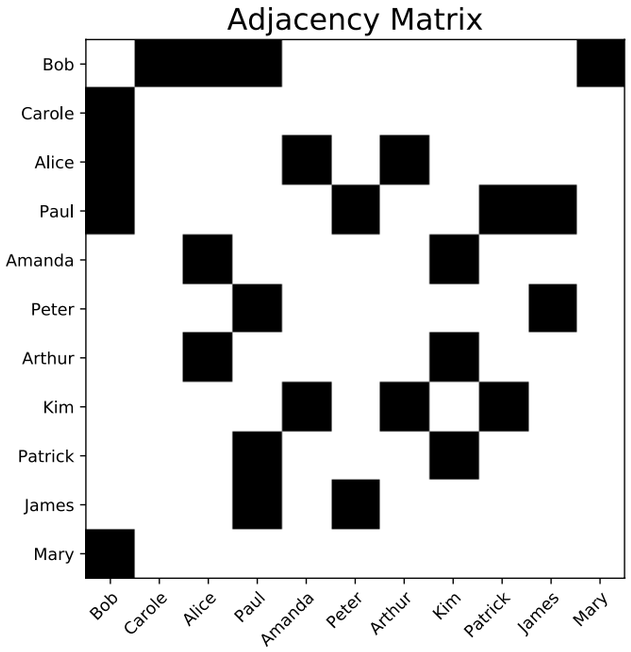




Going Beyond Quantum Chemistry

- By building a general framework, we can apply NTPoly to many different fields.
- Example: computing the matrix exponential $e^{\beta A}$ to study social networks.

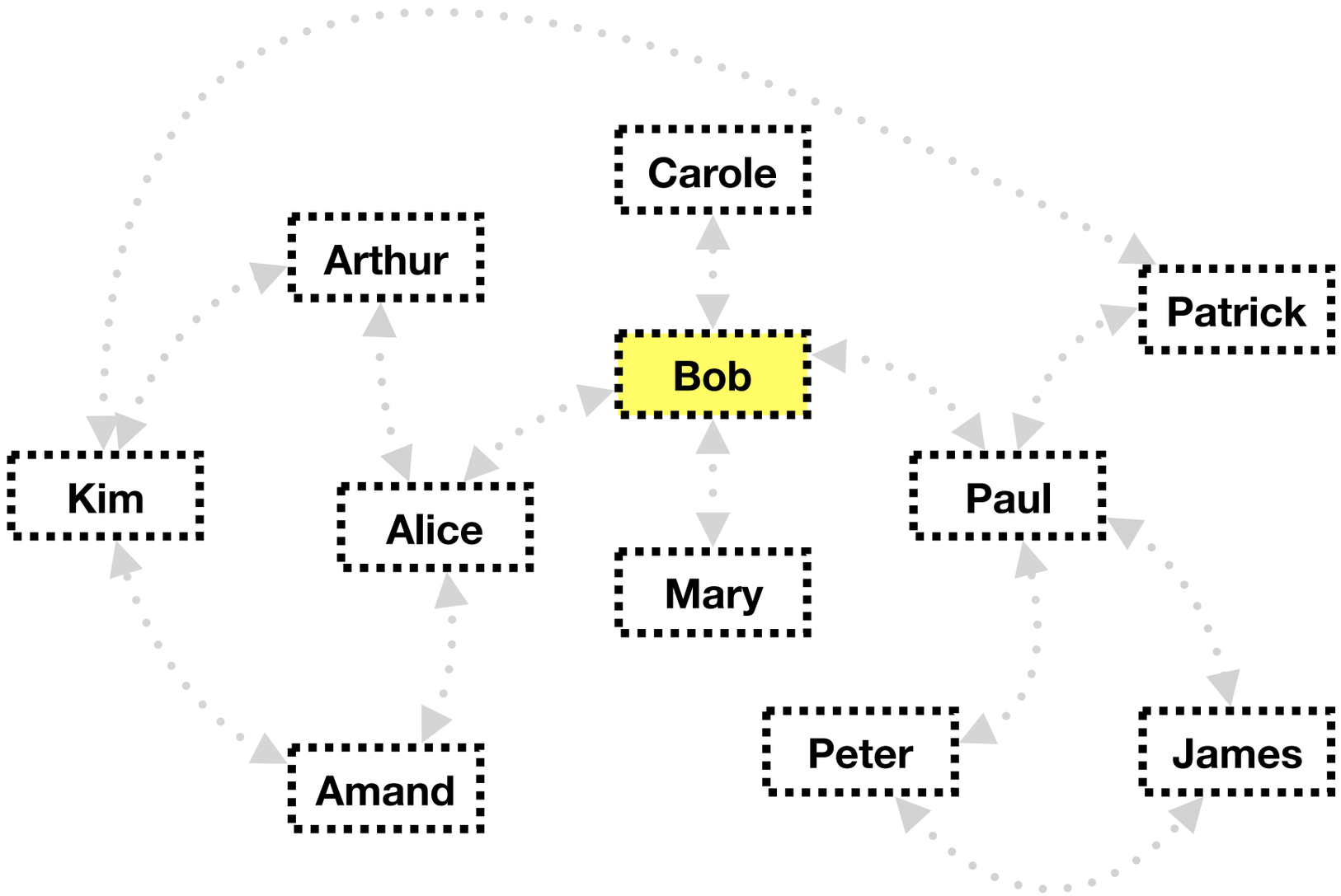


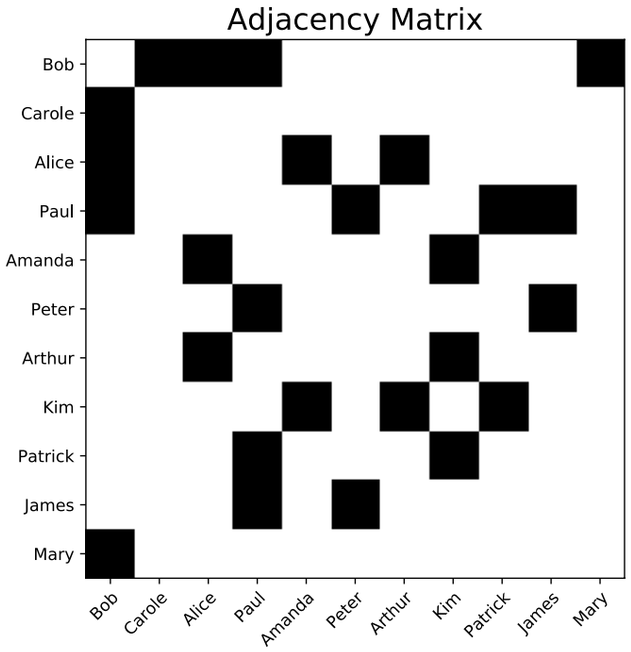


Estrada, Ernesto, Naomichi Hatano, and Michele Benzi. "The physics of communicability in complex networks." Physics reports 514, no. 3 (2012): 89-119.

Going Beyond Quantum Chemistry

- By building a general framework, we can apply NTPoly to many different fields.
- Example: computing the matrix exponential $e^{\beta A}$ to study social networks.

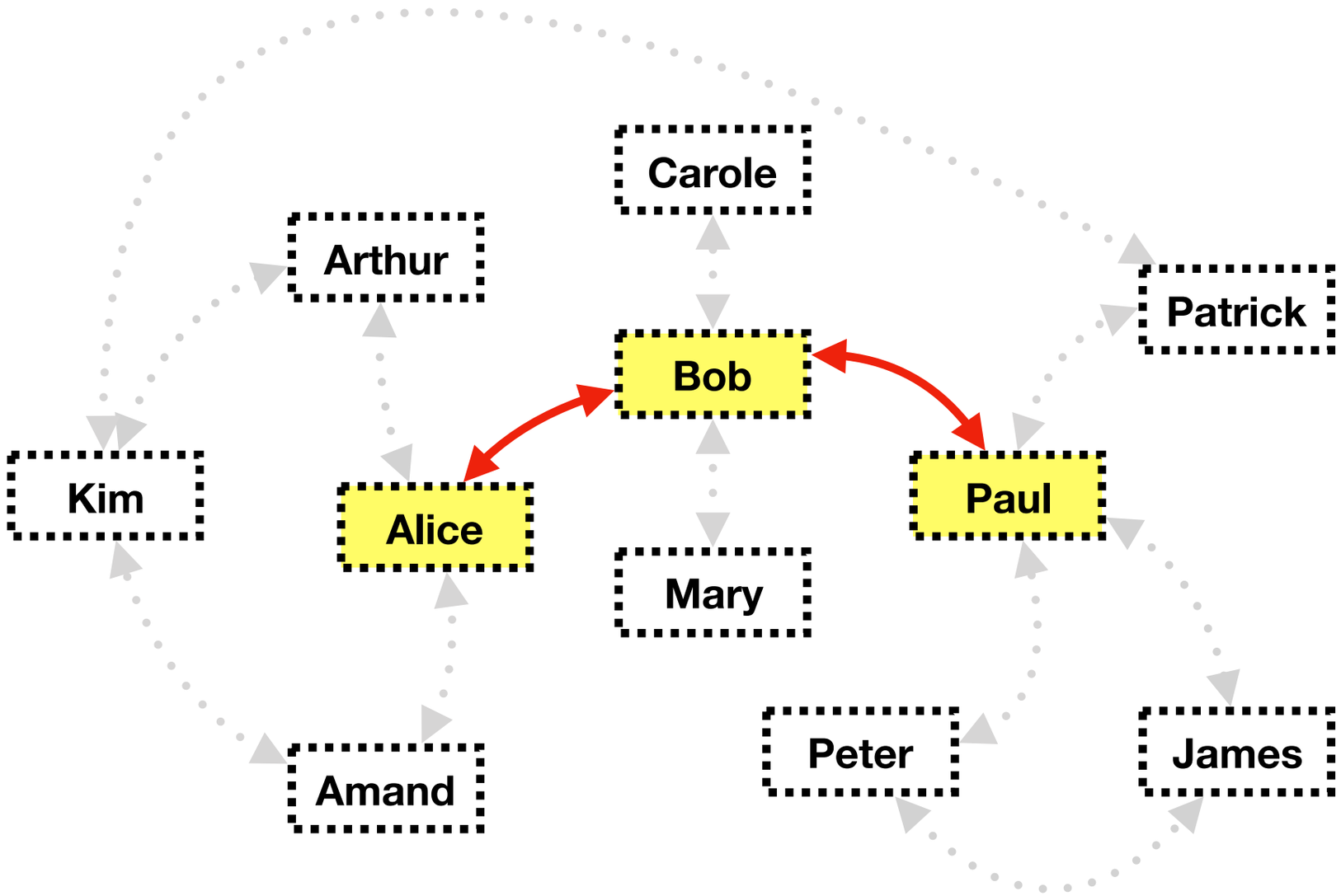


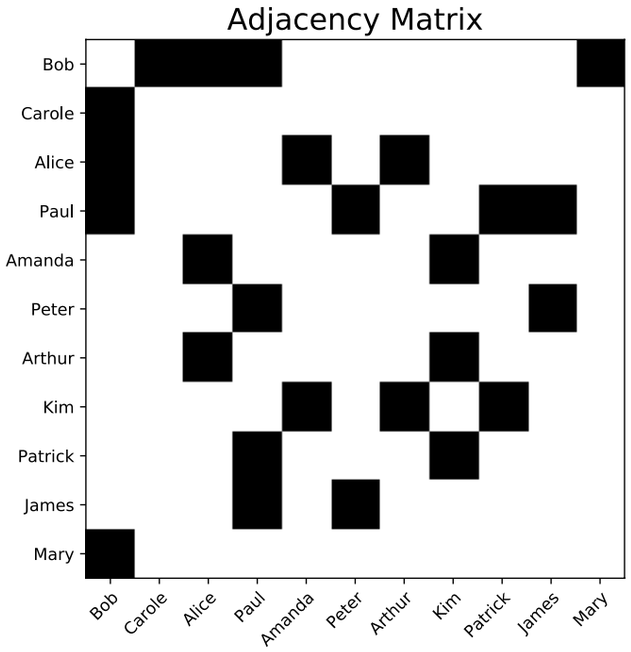


Estrada, Ernesto, Naomichi Hatano, and Michele Benzi. "The physics of communicability in complex networks." Physics reports 514, no. 3 (2012): 89-119.

Going Beyond Quantum Chemistry

- By building a general framework, we can apply NTPoly to many different fields.
- Example: computing the matrix exponential $e^{\beta A}$ to study social networks.

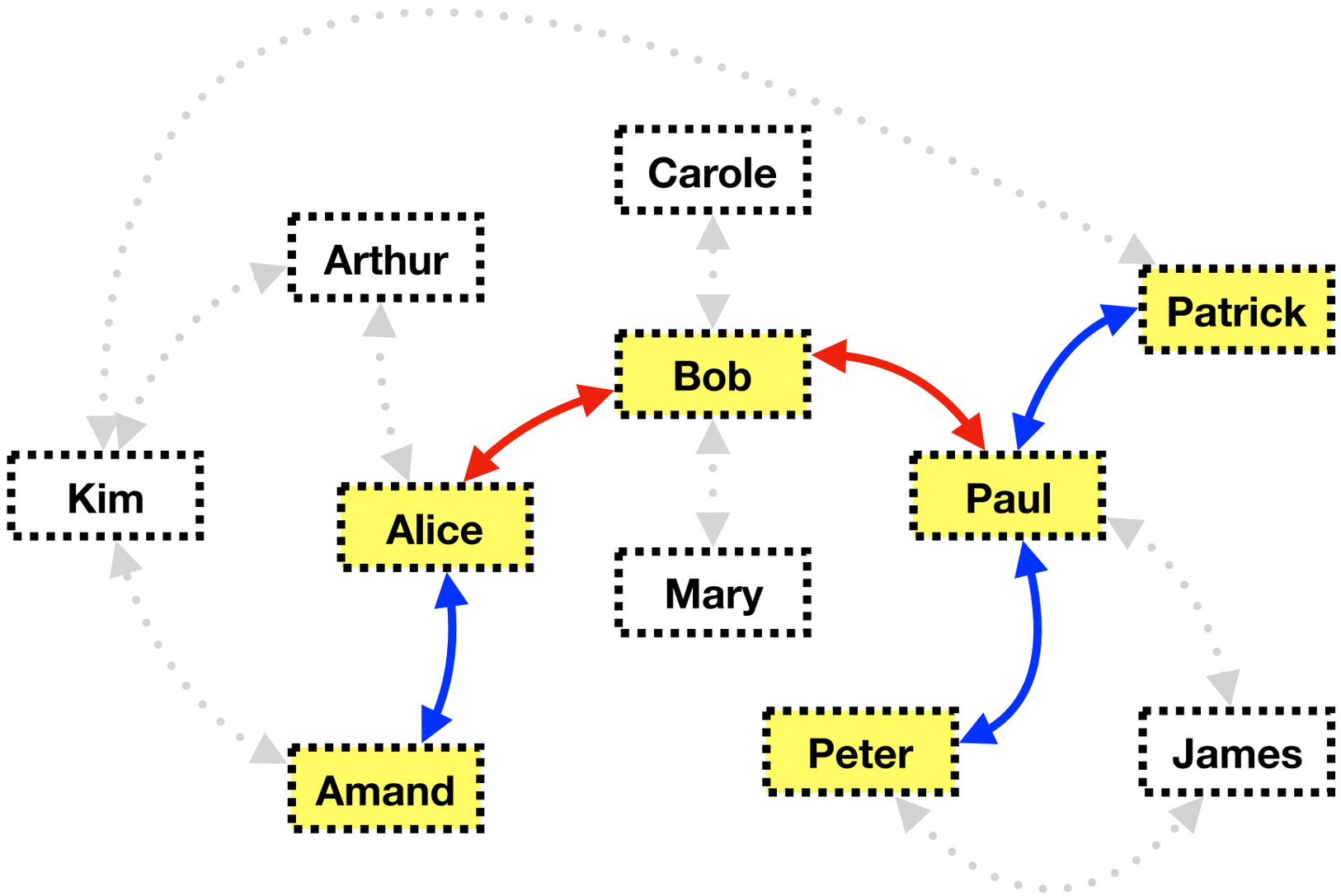


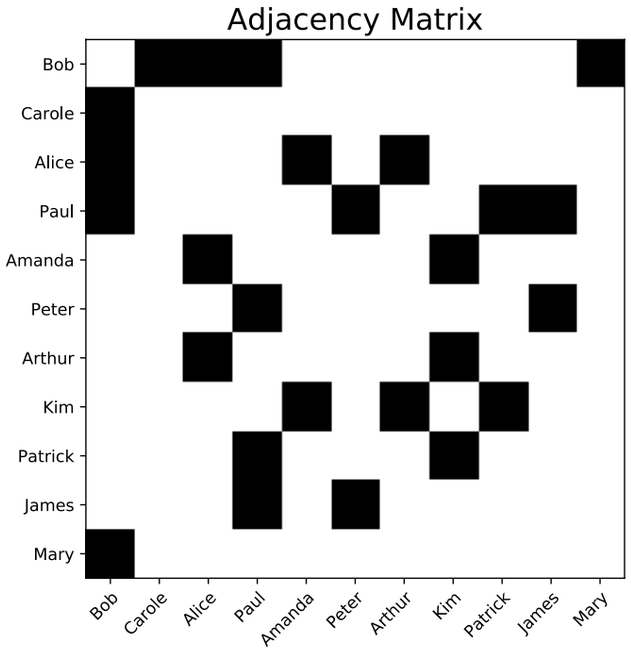


Estrada, Ernesto, Naomichi Hatano, and Michele Benzi. "The physics of communicability in complex networks." Physics reports 514, no. 3 (2012): 89-119.

Going Beyond Quantum Chemistry

- By building a general framework, we can apply NTPoly to many different fields.
- Example: computing the matrix exponential $e^{\beta A}$ to study social networks.

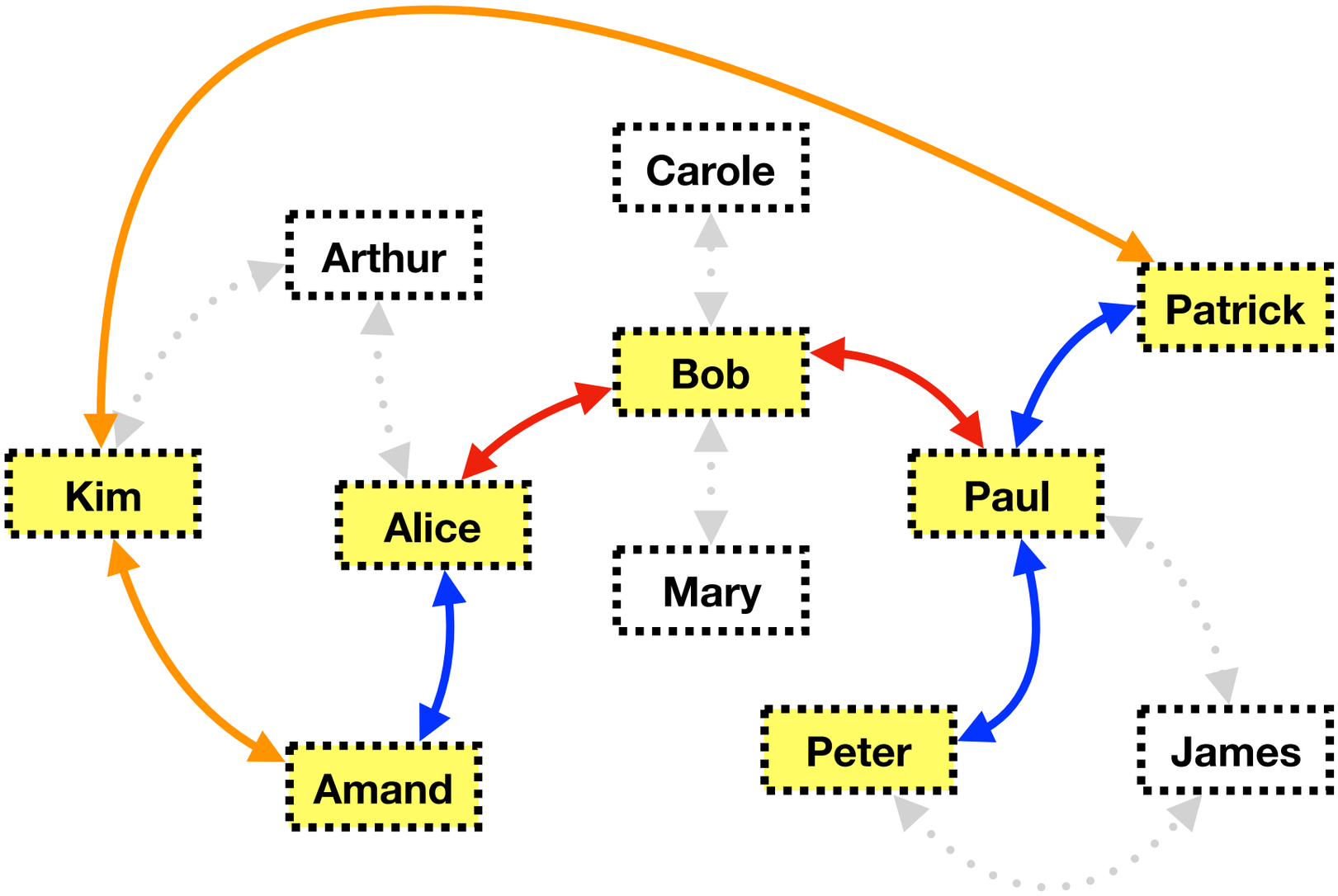


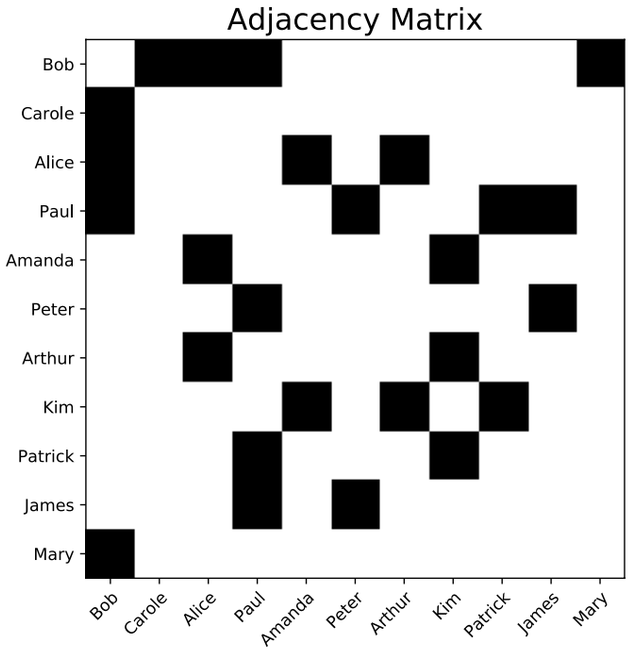


Estrada, Ernesto, Naomichi Hatano, and Michele Benzi. "The physics of communicability in complex networks." Physics reports 514, no. 3 (2012): 89-119.

Going Beyond Quantum Chemistry

- By building a general framework, we can apply NTPoly to many different fields.
- Example: computing the matrix exponential $e^{\beta A}$ to study social networks.

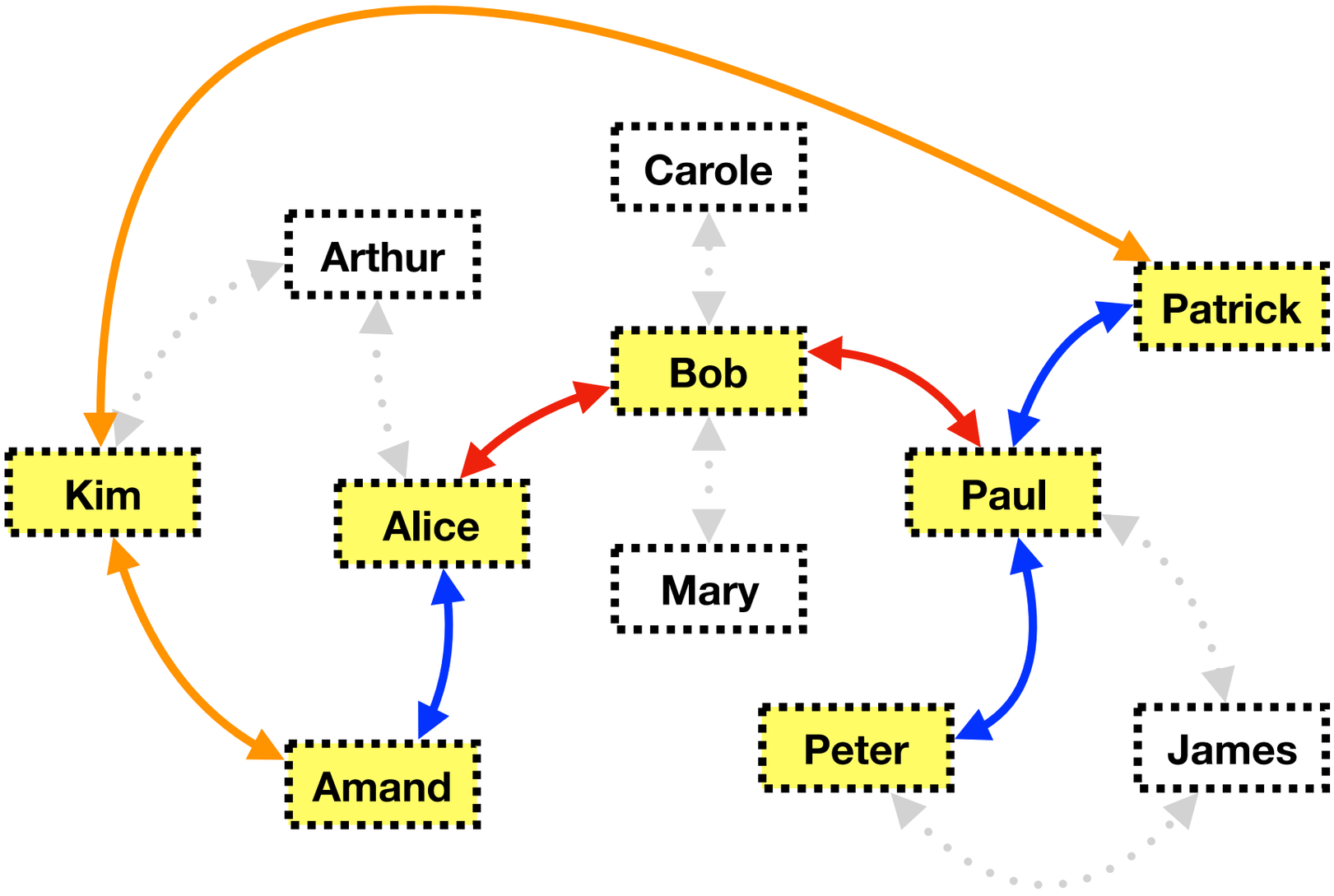


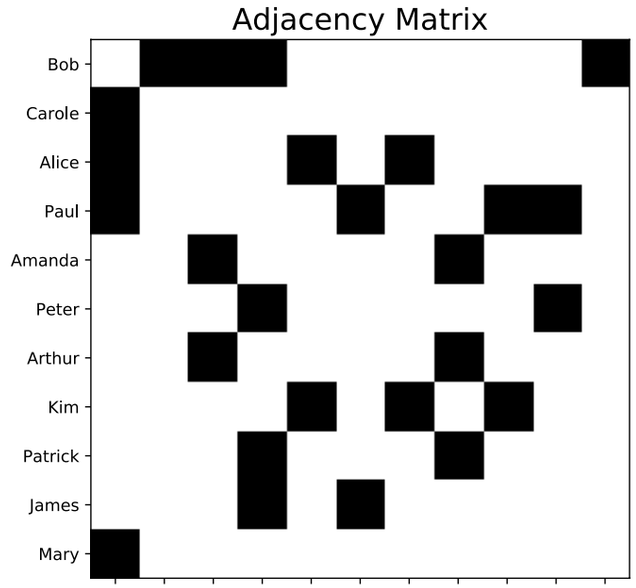


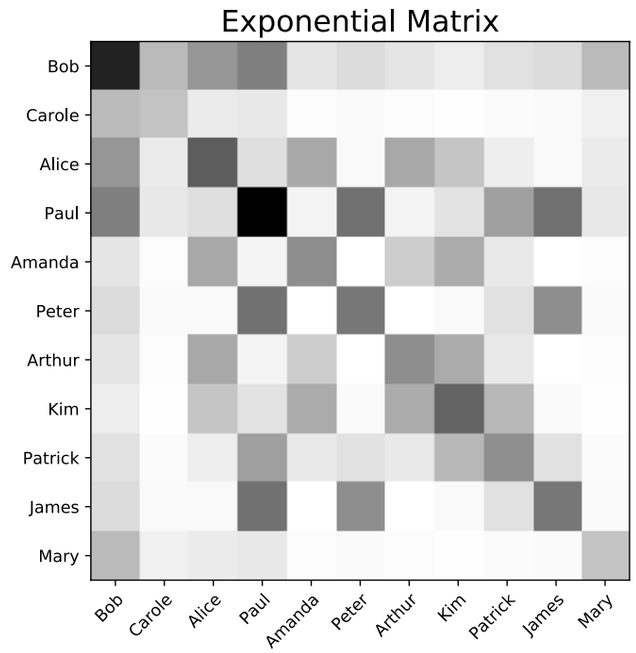
Estrada, Ernesto, Naomichi Hatano, and Michele Benzi. "The physics of communicability in complex networks." Physics reports 514, no. 3 (2012): 89-119.

Going Beyond Quantum Chemistry

- By building a general framework, we can apply NTPoly to many different fields.
- Example: computing the matrix exponential $e^{\beta A}$ to study social networks.



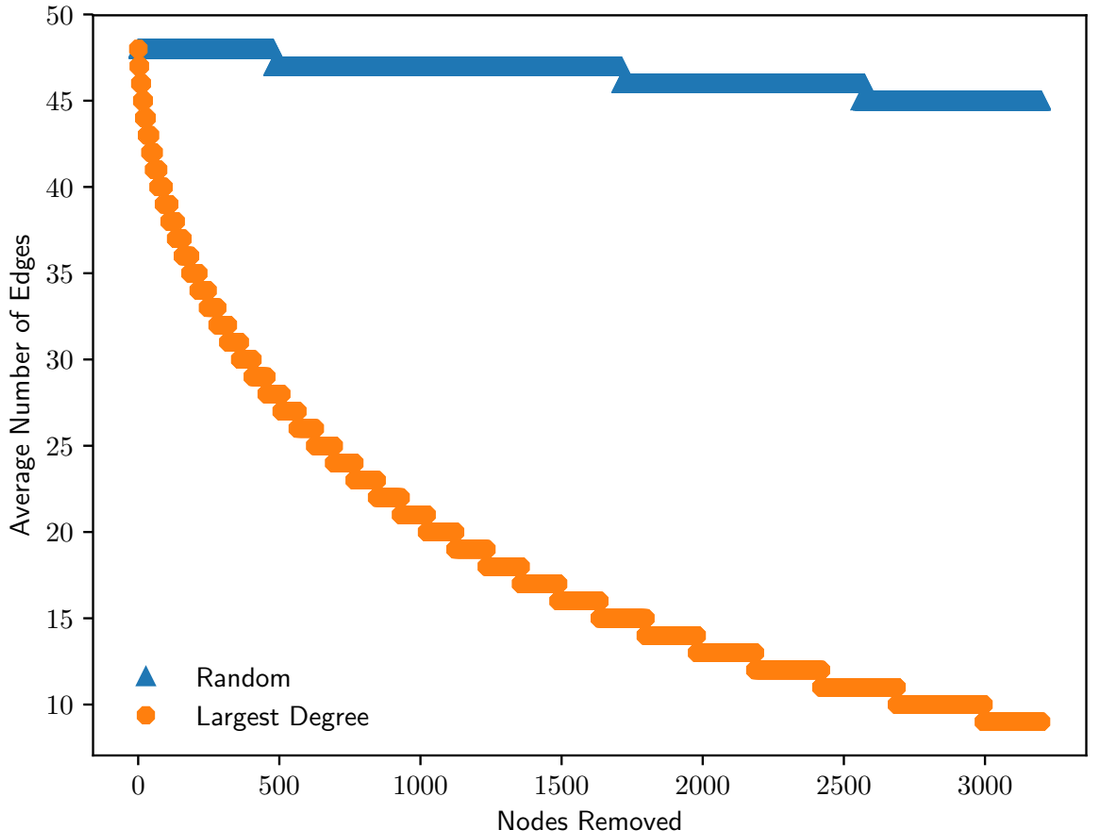




Estrada, Ernesto, Naomichi Hatano, and Michele Benzi. "The physics of communicability in complex networks." Physics reports 514, no. 3 (2012): 89-119.

Example: The Resilience of Social Networks

- Network Resiliency calculations:
  - Data Set: Israeli Social Network “TheMarker Cafe”
  - Nodes: 69413. Sparsity: 0.04%.
- Procedure:
  - Remove a node from the graph
    - Random Node
    - Node with largest degree
  - Compute the matrix exponential
  - Compute the sparsity
  - Repeat



WD, and Takahito Nakajima. "Massively parallel sparse matrix function calculations with NTPoly." *Computer Physics Communications* 225 (2018): 154-165.

Conclusion

- Electronic structure calculations enable us to predict the properties of molecules using the laws of quantum mechanics.
- Using the world's largest supercomputers to push quantum mechanical methods into the 10,000 atom regime, enabling calculations of previously inaccessible systems and properties.
- Use of matrix functions to enable large scale quantum chemistry calculations.
- A highly scalable library that can be applied to many different disciplines.
- Complexity reduction approach to fragment large systems.
- NTPoly: https://github.com/william-dawson/NTPoly
- BigDFT: https://gitlab.com/l_sim/bigdft-suite
- PyBigDFT: https://bigdft-suite.readthedocs.io/projects/PyBigDFT/