Custom Linear Algebra Library for Quantum Chemistry

Sifei Du
University of California, Berkeley
for CHEM 179: Numerical Algorithms Applied to Computational Quantum Chemistry
Prof. Martin Head-Gordon with Austin Glover

Due May 9, 2025

1 Design and Implement Basic Matrix/Vector Classes

The first objective of this project is to build the foundation for all higher-level operations:

- A templated `Matrix<T>` class supporting arbitrary dimensions, dynamic allocation, and row-/column-major storage.
- A `Vector<T>` alias of `Matrix<T>` for $N \times 1$ vectors with convenient element access.
- Operator overloads for addition, subtraction, scalar multiplication, and indexing.

2 Implement Core Linear Algebra Kernels

Extend the library by coding BLAS(Basic Linear Algebra Subprograms)-like routines:

- Matrix–matrix multiplication (`gemm`).
- Matrix–vector multiplication (`gemv`).
- Transpose and trace operations.
- Norms (Frobenius and Euclidean) for error metrics.

3 Implement Symmetric Matrix Diagonalization

Develop an eigen-decomposition routine without external libraries:

- Implement the Jacobi rotation method (Numerical Recipes, Sec. 11.1).
- Validate eigenvalues and eigenvectors on test cases (diagonal matrices, small symmetric examples).

4 Testing and Documentation

Demonstrate the robustness and performance of the library:

- Unit tests covering all routines and edge cases.
- Scaling analysis (compute time vs. matrix size).
- Documentation of the library's API, including usage examples and performance notes.

5 Replace Armadillo in Existing Code (Optional)

Integrate the custom library into the CNDO/2 SCF and extended Hückel codes from Homeworks 3–5:

- Swap out Armadillo matrix types and functions for my own classes and routines.
- Ensure SCF procedures invoke the diagonalizer in place of Armadillo's.
- Validate on H_2 and C_2H_2 test cases, comparing energies and orbitals to reference values.

6 Weekly Milestones

- **Part 1 (April 27 – May 3, 2025):** Design and implement Class/Vector types and core BLAS-like kernels.
- **Part 2 (May 4 – May 7, 2025):** Develop and validate symmetric diagonalization routines (Jacobi and optional QR).
- **Part 3 (May 8 – May 9, 2025):** Perform testing, benchmarking, and finalize documentation.

References

1. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P. *Numerical Recipes: The Art of Scientific Computing*, 3rd ed., Cambridge University Press, 2007.
2. Golub, G.H., Van Loan, C.F. *Matrix Computations*, 4th ed., Johns Hopkins University Press, 2013.
3. Head-Gordon, M.; Glover, A. *C179 Homeworks 3–5*, UC Berkeley, 2025.