

Graduate Research Plan

Zack Morrow

Introduction and Previous Work

At the molecular level, density functional theory (DFT) describes the electronic state of a system through functionals, which are mappings whose domain is a function space [1]. Existing chemistry software packages (e.g., Gaussian) use DFT to provide evaluations of a potential energy surface (PES) at molecular coordinate locations. In order to drive simulations of molecular dynamics, one needs potential energy in order to solve for the reaction path. However, the evaluation of the true PES at the internal coordinates is far too expensive to carry on a dense grid over the full domain.

Sparse grids provide a means to keep the computational cost in check. We compute and store evaluations of the true PES only at the sparse points. We then build a sparse interpolating polynomial of the PES as a surrogate, which we evaluate on the full grid over the domain. The evaluations of the interpolant are significantly less expensive than the evaluations of the true PES; the main expense occurs in evaluating the true PES at the sparse interpolation points, which is done at the front end and stored for later use.

Previous work in our group by James Nance investigated the Smolyak construction of sparse grids and its application to molecular dynamics and surface-hopping problems [2, 3]. Nance worked in collaboration with the research group of Prof. Elena Jakubikova in NC State’s Department of Chemistry, our current collaborators. The sparse grid code that Jakubikova’s group utilizes today is the code written by Nance as part of his thesis.

Proposal: Intellectual Merit

Nature tends to an energy-minimizing state, and accordingly, stable molecular geometries correspond to local minima of the potential energy surface (PES). In order to construct the PES itself, we first take \mathbf{q} to be a fixed geometry and find the electron energy levels \mathcal{E}_i by solving (e.g., with Gaussian) the time-independent Schrödinger equation, a well-known quantum mechanical relationship encoding the energy of a system:

$$\hat{\mathcal{H}}\Psi_i^{\mathbf{q}} = \mathcal{E}_i\Psi_i^{\mathbf{q}}, \quad i \in \{0, 1, 2, \dots\}. \quad (1)$$

Here, $\hat{\mathcal{H}}$ is the molecular Hamiltonian operator (connected to the total energy of the system), the eigenvalue \mathcal{E}_i is the energy of electronic state i under geometry \mathbf{q} , and $\Psi_i^{\mathbf{q}}$ is the wavefunction corresponding to \mathcal{E}_i (connected to the probability that a given electron in a molecule under geometry \mathbf{q} has energy \mathcal{E}_i).

Now, as a function of all admissible geometries \mathbf{q} , we denote $\mathcal{E}_i(\mathbf{q})$ to be the PES corresponding to energy state i . Again, stable molecular geometries correspond to local minima on the PES; moreover, transitions of a molecule from energy state $i + 1$ to state i occur when the molecular geometry corresponds to a local minimum on $\mathcal{E}_{i+1}(\mathbf{q})$. The reaction path to an energy-minimizing configuration is the solution to

$$\dot{\mathbf{q}} = -\nabla\mathcal{E}_i(\mathbf{q}), \quad (2)$$

which, pictorially, is the path of steepest descent on the PES from the initial geometry to a local minimum. Using Equation (2), we can simulate numerically how molecular geometry evolves in time after excitation to higher energy states and subsequent relaxation.

Since the true PES $\mathcal{E}_i(\mathbf{q})$ is far too expensive to compute on the full grid, we generate a set of sparse grid points and utilize the sparse interpolating polynomial $\mathcal{E}_i^s(\mathbf{q})$ as a surrogate for the true PES in our simulations. Currently, we use a MATLAB code to manage the generation and evaluation of the sparse interpolant $\mathcal{E}_i^s(\mathbf{q})$, and the dynamical simulation routines run on university desktops and laptops. However, the simulation code will soon be migrating to the XSEDE platform, a geographically distributed computing cluster on which it is infeasible to use MATLAB. We are therefore replacing the MATLAB sparse-grid manager with an open-source C++ sparse-grid manager called Tasmanian, developed at Oak Ridge National Laboratory, which additionally comes with a Python wrapper [5].

I have replicated previous PES results using the new Tasmanian package and am currently working to integrate it fully into our dynamical simulation codes. One difficulty to resolve is that Tasmanian currently does not compute gradients internally—a problem when using $\mathcal{E}_i^s(\mathbf{q})$ in Equation (2). To boost accuracy and exploit parallelizability, gradients are best evaluated inside the sparse-grid manager itself. I will spend next summer at Oak Ridge working to add gradient-computation routines to the Tasmanian package, which must meet rigorous Department of Energy software standards. In the meantime, I am familiarizing myself with the high-performance computing environment at NC State, in addition to becoming as comfortable with C++ as I am with Python and MATLAB. Beyond next summer, I will very likely need to refine the gradient routines, and I will also continue my literature review on quantum chemistry and the analysis of sparse grids.

Proposal: Broader Impacts

Sparse grids as a general category have ready applications in any field where computational cost is a major concern. In the realm of chemistry, computationally tractable methods of handling high-dimensional DFT-driven dynamical simulations have a wide array of benefits to society, including public health, renewable energy, and national security. The research I propose to undertake can advance the state of the art of simulations in pharmacological modeling, photochemistry, conversion in solar cells, nuclear chemistry, and nuclear power [3, 4], to name a few. Additionally, the gradient routines that I will add to the Tasmanian code will be useful to all who use the publicly available Tasmanian package to manage sparse grids in their research.

Conclusion

Sparse grids are powerful tools, making simulations that were once intractable become feasible. I possess the drive, prior computational laboratory experience, and proven academic background to carry this project through to completion and to communicate my results to both specialist and non-specialist audiences.

- [1] Hohenberg, P. and Kohn, W. “Inhomogeneous Electron Gas”. *Phys. Rev.* 136.3B (1964), B864–B871.
- [2] Nance, J. and Kelley, C. T. “A Sparse Interpolation Algorithm for Dynamical Simulations in Computational Chemistry”. *SIAM J. Sci. Comput.* 37.5 (2015), S137–S156.
- [3] Nance, James D. “Investigating Molecular Dynamics with Sparse Grid Surrogate Models”. PhD thesis. North Carolina State University, 2015.
- [4] Peherstorfer, Benjamin et al. “Selected Recent Applications of Sparse Grids”. *Numerical Mathematics: Theory, Methods, & Applications* 8.1 (2015), pp. 47–77.
- [5] Stoyanov, M. *TASMANIAN Sparse Grids*. Tech. rep. ORNL/TM-2015/596. ORNL, 2015.