Scaling Quantum Mechanics and First Principles Dynamics for Accuracy and Efficiency

Todd J. Martínez
SLAC National Accelerator Laboratory
Stanford University

SciDAC4 Project: Designing Photocatalysts Through Scalable Quantum Mechanics and Dynamics
Project Overview

• Improve computational modeling tools for modern architectures to enable design of improved photocatalysts

• Major Thrusts
  – Enable Rapid Prototyping and deployment of multi-level parallel algorithms (Alex Aiken, Kunle Olukotun, Lexing Ying)
    • Legion/Regent for high-level parallel expression
    • DeLite to build domain specific languages for quantum chemistry and first principle dynamics
  – Develop and implement modular library frameworks for large-scale atomistic simulations (Ed Hohenstein, Todd Martínez, Robert Parrish)
    • Lightspeed framework to build electronic structure from highly tuned primitives
  – Design of photoactivated enzymes (Ron Dror, Possu Huang, TJ Lane, Henry van den Bedem)
    • Protein design with novel cofactors
    • QM/MM investigations of photoenzymes
FAP: CO$_2$ Photochemistry in Nature

Fatty Acid Photodecarboxylase

- Photo-activated production of alkanes
- Requirement for light: not yet clear
- Mechanism: not known
- **Project goal**: QM calculations to predict mechanism, verify by comparing to spectroscopy, crystallography

\[
\begin{align*}
\text{hv} & \quad \rightarrow \\
\left(\text{alkane}\right)_n & \quad + \quad \text{CO}_2
\end{align*}
\]
Like all photoenzymes, FAP contains an “antenna” molecule (cofactor) that absorbs photons and plays a key role in the chemistry.

QM/MM calculations of excited state mechanism are in progress.

Large QM regions: Need fast and accurate QM!

See Poster by TJ Lane
Legion

• Task-based programming model
  – Heterogeneous machines
  – Distributed memory

• Key features
  – Programs are written without specifying where computations will run and where data will be placed
  – Separate *mapping* phase allows program to be tuned to a particular machine

• Currently working to apply Legion to electron repulsion integral generation

![Task graph for one time step of a simple application]
DeLite

• DeLite framework allows for easy specification of domain specific languages
  – Domain specific optimizations
  – Structure computation and data
  – Optimize mapping to different hardware targets such as GPUs or FPGAs

• Working to develop DSL for J/K matrix build

• Using DeLite multiscale neural network DSL to machine learn better exchange-correlation functionals

\[
\left[-\frac{1}{2} \nabla^2 + V_{KS}[\rho] \right] \psi_i(r) = \lambda_i \psi_i(r), \quad \rho(r) = \sum_{i=1}^{N_e} |\psi_i(r)|^2
\]

Learn the nonlinear map from potential to density
Primitives for Electronic Structure?

SCF Energy

\[ J^{(D)}_{\mu \nu} = \sum_{\rho \sigma} (\mu \nu | \rho \sigma) D^{\rho \sigma} \]

Electron Repulsion Integrals (ERIs)

\[ K^{(D)}_{\mu \nu} = \sum_{\rho \sigma} (\mu \rho | \nu \sigma) D^{\rho \sigma} \]

SCF Gradients

\[ J(A, B)^{\xi} = \sum_{\mu \nu \rho \sigma} (\mu \nu | \rho \sigma)^{\xi} A^{\mu \nu} B^{\rho \sigma} \]

\[ K(A, B)^{\xi} = \sum_{\mu \nu \rho \sigma} (\mu \rho | \nu \sigma)^{\xi} A^{\mu \nu} B^{\rho \sigma} \]

Density Matrix

Continuum Solvation and QM/MM

\[ V_k = \sum_{\mu \nu} P_{\mu \nu} (\mu \nu | k) \]

\[ \Delta F^S_{\mu \nu} = \sum_k q_k (\mu \nu | k) \]

We built fast GPU versions of these primitives
Build a framework that allows easy reuse?
R.M. Parrish, T.J. Martinez and co-workers, in preparation.
import lightspeed as ls # The lightspeed module
resources = ls.ResourceList.build() # Use all available CPU/GPU resources
molecule = ls.Molecule.from_xyz_file('geom.xyz') # Read ./geom.xyz and build Molecule
basis = ls.BasisSet.from_gbs_file(molecule, 'cc-pvdz') # Construct cc-pVDZ basis
pairlist = ls.PairList.build_schwarz(basis, basis, 1.0E-14) # Construct Pairlist
S = ls.IntBox.compute_overlap(resources, pairlist) # Compute the overlap matrix as Tensor
J = ls.IntBox.compute_coulomb( # Compute the Coulomb matrix as Tensor
    resources, # The resources to use
    ls.Ewald.coulomb(), # The standard Coulomb interaction operator
    pairlist, # The pairlist on the bra (12|
    pairlist, # The pairlist on the ket |34)
    S, # The input density matrix (S used for demo only)
    1.0E-6, # The double-precision cutoff
    1.0E-14) # The single-precision cutoff
print J
import lightspeed as ls  # The lightspeed module
import psiw  # The "psidewinder" lightweight electronic structure module
import md  # The lightweight adiabatic MD code
# CPU and/or GPU resources
resources = ls.ResourceList.build()
# OpenMM-based QM/MM (Mechanical + Coulomb embedding w/ link H atoms)
qmmm = psiw.QMMM.from_prmtop(
    prmtopfile='pyp.prmtop',
    inpcrdfile='pyp.rst',
    qmindsfile='pyp.qm',
    charge=-1.0,
)
# Geometry manages all external environment considerations
geom = psiw.Geometry.build(
    resources=resources,
    qmmm=qmmm,
    basisname='6-31g',
)
# FON-RHF (4 active electrons in 3 fractional orbitals)
ref = psiw.RHF.from_options(
    geometry=geom,
    g_convergence=1.0E-6,
    fomo=True,
    fomo_method='gaussian',
    fomo_temp=0.2,
    fomo_nocc=107,
    fomo_nact=3,
)
Lightspeed Example: QM/MM FOMO-CASCI (continued)

ref.compute_energy()
# FOMO-CASCI (3 singlet states)
casci = psiw.CASCI.from_options(
    reference=ref,
    nocc=107,
    nact=3,
    nalp=2,
    nbet=2,
    S_ind=[0],
    S_nstate=[3],
)
casci.compute_energy()

Key Methods:

grad = cas.compute_gradient(0, 0)
coup = cas.compute_coupling(0, 0, 1)
overlap = psiw.CASCI.compute_overlap(cas, cas, 0)

And another dozen lines to production-scale AIMD (see Parrish poster)...
Lightspeed Enables Rapid Prototyping: Parallax - Neglect of Fragment Differential Overlap (NFDO)

Current State of the Art:

At the Least, Coulomb Embedding is Required:

$E \equiv \sum_A E_A^{SCF}$

$+ \frac{1}{2} \sum_A \sum_{B'} E_{AB'}^{Coulomb}$

$+ \frac{1}{2} \sum_A \sum_{B'} E_{AB'}^{6-12}$

Next Target:

Large Gas-Phase

Large PBC
Parallax: PME Approach

Raw Density: \( \rho(\vec{r}_1) \)

Gaussian-Blurred Density: \( \tilde{\rho}(\vec{r}_1) \)

ESP of blurred density (long-range Ewald ESP):

\[
\tilde{\rho}(\vec{r}_1) \xrightarrow{\text{FFT}} \hat{\rho}(\vec{k}) \xrightarrow{4\pi/\vec{k}^2} \hat{Z}(\vec{k}) \xrightarrow{\text{IFFT}} \tilde{Z}(\vec{r}_1)
\]

Parallax: Full Coulomb Embedding (Long Range)
Parallax: Full Coulomb Embedding (Short Range)

$\langle pq | \text{erfc}(\omega r_{12})/r_{12} | rs \rangle$

$\Rightarrow$ Linear Scaling, highly but not embarrassingly parallel.
**Parallax: Sensitivity to Grid/Ewald Parameters**

Short-Range Cutoff Distance:

| $|\Delta E|$ [a.u.] | $R$ [a.u.] |
|----------------|----------|
| $10^0$       | 0        |
| $10^{-1}$    | 1        |
| $10^{-2}$    | 2        |
| $10^{-3}$    | 3        |
| $10^{-4}$    | 4        |
| $10^{-5}$    | 5        |
| $10^{-6}$    | 6        |
| $10^{-7}$    | 7        |
| $10^{-8}$    | 8        |
| $10^{-9}$    | 9        |
| $10^{-10}$   | 10       |

$\omega = 0.75$ a.u.

Fourier Grid Spacing:

| $|\Delta E|$ [a.u.] | $|\Delta x|$ [a.u.] |
|----------------|-------------------|
| $10^0$       | 0.4               |
| $10^{-1}$    | 0.5               |
| $10^{-2}$    | 0.6               |
| $10^{-3}$    | 0.7               |
| $10^{-4}$    | 0.8               |
| $10^{-5}$    | 0.9               |

Parallax: RHF/STO-3G
648-Atom Unit Cell
$-1.6 \times 10^4$ E$_h$ SCF Energy
Parallax: Scaling for Simple Water Box

Parallax: RHF/STO-3G Timings on PBC Water Boxes (10 SCF Iterations + Gradient)

Wall Time [h]

Grid Collocation
FFT
Short-Range Coulomb
Exchange
Gradient
Other

N=1
N=2
N=4

$N_{atom} = 648N^3$

12-Core Intel E5-2640@2.5 GHz/400 GB RAM
Parallax – Next Steps

• Benchmark accuracy of approach
• Application to chemical reactions in solution
  – Reacting system treated as a single fragment
• Extend to allow for fragmentation across covalent bonds
• Extend to allow for dynamic fragmentation
Recommendation Systems

How Companies Learn Your Secrets

By CHARLES DUHIGG
Published: February 16, 2012 | 570 Comments

“Alexa, play music everywhere.”
## Recommendation Systems

<table>
<thead>
<tr>
<th></th>
<th>Dark City</th>
<th>Star Wars</th>
<th>Zero Dark Thirty</th>
<th>Steel Magnolias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe</td>
<td>1</td>
<td>?</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Fred</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Jane</td>
<td>?</td>
<td>?</td>
<td>5</td>
<td>?</td>
</tr>
<tr>
<td>Alex</td>
<td>?</td>
<td>3</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Sara</td>
<td>1</td>
<td>?</td>
<td>?</td>
<td>5</td>
</tr>
</tbody>
</table>

Need to complete the matrix…
Is this possible?
   Only if the matrix does not have much information!

Machine learning strategy: ASSUME the problem is well-posed

In example above – 10 numbers out of 20, assume this is enough to determine remaining 10 numbers…
Singular Value Decomposition

\[ A = U \Sigma V \]

Diagonal
Rank = number nonzero diagonal elements

Assume rank is small and find best low rank approximation that reproduces known entries in the matrix...
Full CI

Full Configuration Interaction:

\[ \psi = \sum_{I} c_I |I\rangle \]

All possible arrangements of N electrons in M orbitals

\[ c_1 \phi_1 \phi_1 + c_2 \phi_1 \phi_2 + c_3 \phi_2 \phi_1 + c_4 \phi_2 \phi_2 \]

Need matrix-vector products \( \mathbf{Hc} \), vector length \( N_I \) is \textit{factorial} in N,M

Formal cost:

\[ O\left( N_I^2 \right) \]

Accounting for sparsity of \( \mathbf{H} \):

\[ O\left( N_I \right) \]
Rank Sparsity in Full Cl?

How can we reveal noninformative nature of Cl vector?

Rearrange vector to a matrix:

\[
\left\{ \varphi_1, \varphi_2 \right\} \otimes \left\{ \overline{\varphi}_1, \overline{\varphi}_2 \right\}
\]

“α-string” “β-string”

\[
\psi = \sum_{IJ} C_{IJ} |\alpha_I \beta_J \rangle
\]

Now, play same trick from recommendation systems:

\[
C = \sum_r^{N_r} \lambda_r P_r Q_r^T
\]

Memory requirement, Computational cost: \(O\left(\sqrt{N_I}\right)\)

Note early work by Koch and later also Taylor...
Are Electronic Wavefunctions Informative?

50 terms (out of $10^4$) are enough for kcal/mol accuracy!

Even fewer terms needed for accurate energy differences!
rr-FCI Performance

Even the “conventional” FCI timings here are FAR faster than expected – this is because of Fales-Levine work on GPU-based Full CI

Number of grains of sand on Earth: $10^{18}$

Age of the universe: $10^{17}$ sec

$10^{16}$ determinants ≈ $10^{15}$ seconds w/ conventional FCI

$O(N^{0.543})$

$O(N^{1.079})$

≈ $10^{10}$ determinants!
rr-FCI

- Renaissance in selected CI methods – exploiting element sparsity of CI vector (Evangelista, Head-Gordon, Umrigar, Alavi): FCI-QMC, Heat Bath-CI, Adaptive Sampling CI
- These also increase the efficiency of full CI – some are deterministic (ASCI) and others are stochastic (FCI-QMC, HBCI)
- rr-FCI exploits *rank* sparsity of CI vector
  - There is an rr-FCI value for *every* element of the CI vector
  - This is qualitatively different from approximations which neglect small elements
- Deterministic, so gradient is straightforward for dynamics
- Working to improve convergence and to exploit mixed precision
Tensor Hypercontraction

Applying decomposition techniques to 4\textsuperscript{th} order tensor:

\[(ij \mid kl) = \sum_{PQ} X_{ip}X_{jP}Z_{PQ}X_{kQ}X_{lQ}\]

P,Q indices have approx. same range as basis

i,j,k,l are \textit{unpinned}! Can sum over any of these without carrying along the others...

Numerical methods to determine X and Z simultaneously...
Analytic formulas to determine Z, given X
Numerical methods to determine X, given Z

Formal scaling of all pair methods reduced to O(N\textsuperscript{4})!
Implementation of THC

\[ E_{\text{MP2}} = \sum_{ijab} t_{ij}^{ab} \left[ 2(ia|jb) - (ib|ja) \right] \]

\[ E_{\text{THC-MP2}} = \sum_\nu \sum_{ijab} \sum_{PQRS} \tau_\nu^i \tau_\nu^j \tau_\nu^a \tau_\nu^b \cdot X_i^P X_a^P Z_i^{PQ} X_j^Q X_b^Q \times \]

\[ [2 \cdot X_i^R X_a^R Z_i^{RS} X_j^S X_b^S - X_i^R X_b^R Z_i^{RS} X_j^S X_a^S] \]

Graphical Representation of J-like term

Where to put parenthesis?
Automatic Factorization for THC

\[
E \leftarrow 2 \sum_{\nu QR} \left( \sum_P \left( \sum_i \tau_i^\nu X_i^P X_i^R \right) \left( \sum_a \tau_a^\nu X_a^P X_a^R \right) Z^{PQ} \right) \left( \sum_S \left( \sum_j \tau_j^\nu X_j^Q X_j^S \right) \left( \sum_b \tau_b^\nu X_b^Q X_b^S \right) Z^{RS} \right)
\]
Conclusions

• Lightspeed framework being developed to rapidly prototype new algorithms: will provide lessons for more flexible DSLs and implementation of core “boxes” in Legion

• NFDO method being developed using the Lightspeed framework – demonstrates ability to rapidly prototype in this environment

• Rank reduced Full CI leverages low complexity of CI wavefunctions

• Rank reduction techniques can also be used for electron repulsion integrals with tensor hypercontraction – we are exploring automated code generation and optimization to implement THC algorithms

• Simulations of photoenzyme activity in FAP are underway and will be enabled/facilitated by quantum chemistry and first principles dynamics developments
Acknowledgments

- Alex Aiken
- Kunle Olukotun
- Lexing Ying
- TJ Lane
- Henry Van Den Bedem
- Ron Dror
- Possu Huang
- Ed Hohenstein
- Robert Parrish
- Alice Walker
- Scott Fales
- Henrik Koch
- Stefan Seritan
- Benjamin Levine
- Nick Settje

$: DOE ASCR / DOE BES