S4490 - Moving Biophysics to the GPU Cloud for Studying Energy-Transfer in Photosynthesis

Tobias Kramer in collaboration with Christoph Kreisbeck

Institute of Physics, Humboldt-Universität Berlin, Germany
Dept. of Chemistry and Chemical Biology, Harvard University, USA
Schrödinger’s cat alive or dead? Quantum mechanics in photosynthesis

**bio system** atomistic model of photosynthetic complex available?
- ✓ Fenna-Matthews-Olson (FMO) complex, $3 \times (7-8)$ bacteriochlorophylls

**experiment** laser pulse triggers excitation $\rightarrow$ track energy transport
- ✓ Femtosecond pump-probe experiments on FMO

**evidence** quantum-mechanical superposition seen?
- $\rightarrow$ interpretation of experimental signal requires theory
- ✓ high-performance GPU solution (this talk)

**theory** accurate numerics for the biological relevant parameter space tedious!

**relevance** learning design lessons from nature for synthesis of organic solar cells

---

3 Engel et al.: Nature, 2007, **446**, 782
Both alive and dead: laser experiments

- **Delay time**: 0 femto seconds (fs)
- **Delay time**: 280 fs
- **Delay time**: 155 fs
- **Delay time**: 600 fs

▶ pump photosynthetic Fenna-Matthews-Olson complex with laser pulse and probe at later time

→ white-arrow indicates oscillating “blob” (superposition of eigenstates)

---

Not alive: simple theoretical model

Excitonic from antenna to reaction center via $d$ sites:
rate equation for population transfer

\[ \frac{d}{dt} \rho_{ii}(t) = \sum_{m=1}^{d} k_{im} [\rho_{mm}(t) - \rho_{ii}(t)] \]

Rate $k_{im}$ temperature dependent, given by Förster resonant energy-transfer for network Hamiltonian $\mathcal{H}$

Time-dependent $t$ solution by exponentiation:

\[ \rho(t) = \rho(0) \exp[kt] \]

Easily solvable by matrix diagonalization.
No oscillatory behaviour, long-time thermal state.

---

1. Pearlstein, Photochem Photobiol, 1982, 35, 835
2. Frigaard et al., J Bacteriology, 2002, 184, 3368
The Frenkel-exciton model of energy-transfer

![Diagram of FMO complex with BChls](image)

\[ H = \sum_{m=1}^{d} \epsilon_{m} |m\rangle \langle m| + \sum_{m > n} J_{mn} (|m\rangle \langle n| + |n\rangle \langle m|) + \sum_{m,i} |m\rangle \langle m| \lambda_i (b_{i,m}^\dagger + b_{i,m}) + \sum_{m,i} \hbar \omega_i b_{i,m}^\dagger b_i \]

Exciton dynamics (7 x 7 matrix) + Vibrational coupling

The FMO complex transfer light-induced excitons (electron-hole pairs) from the antenna to the reaction centre (RC), where a chemical reaction stores the energy.

Choosing a scalable computational method

Typical “physics problem”: different ways to obtain solution.

Which method works well on GPU?

- Monte-Carlo evaluation, requires lots of trajectories
- quasi-adiabatic path integral method (QUAPI)
- ...
- “hierarchical equations of motion” (HEOM), considered to be too slow

Combined knowledge of physics and computer science required to pick scalable algorithm.
Hierarchical equations of motion (HEOM): formalism

\[
\frac{d}{dt} \rho(t) = -\frac{i}{\hbar} \left[ \mathcal{H}, \rho(t) \right] + \sum_{m=1}^{d} i V_{m,\text{vibr}} \sigma^{P_m\{1,\ldots,0\}}(t) \]

coherent dynamics

\[
\sum_{m=1}^{d} n_m \gamma \sigma^{(n_1,\ldots,n_d)}(t)
\]

coupling to vibrations

millions of auxiliary density matrices \( \sigma^{(m_1,\ldots,n_d)}(t) \) store vibrational state

\[
\frac{d}{dt} \sigma^{(n_1,\ldots,n_d)}(t) = -\frac{i}{\hbar} \left[ \mathcal{H}, \sigma^{(n_1,\ldots,n_d)}(t) \right] + \sum_{m=1}^{d} n_m \gamma \sigma^{(n_1,\ldots,n_d)}(t)
\]

small matrix × small matrices (aligned)

\[
+ \sum_{m=1}^{d} i V_{m,\text{vibr}} \sigma^{(n_1,\ldots,n_m+1,\ldots,n_d)}(t)
\]

small matrix × small matrix (jumps)

\[
+ \sum_{m=1}^{d} n_m \theta_m(\gamma) \sigma^{(n_1,\ldots,n_m-1,\ldots,n_d)}(t)
\]

small matrix × small matrix (jumps)

\[\gamma\text{ dissipative vibrational mode density } J(\omega) = \frac{2\gamma \lambda \omega}{\gamma^2 + \omega^2}\]

\[\gamma\text{ hierarchy truncated at } N_{\text{max}} = \sum_{m=1}^{d} n_m \text{ (typical } N_{\text{max}} \sim 10)\]

---

The challenge: implement efficient algorithms

Quantum dynamics described in terms of a hierarchy of time-dependent differential equations with d-tuples \( \vec{n} = (n_1, \ldots, n_d) \), \( n_i \in \mathbb{N}^0 \):

\[
\frac{d}{dt} \sigma^{\vec{n}}(t) = -\frac{i}{\hbar} [\mathcal{H}_i, \sigma^{\vec{n}}(t)] + \sum A_i \sigma^{\vec{n}^-}(t) + B_i \sigma^{\vec{n}^+}(t)
\]

Coupling: \( \sigma^{\vec{n}^-} = \sigma^{\vec{n}^-}(...,1,...) \) and \( \sigma^{\vec{n}^+} = \sigma^{\vec{n}^+}(...,1,...) \)

Example of 4-tuples sorted by termination level \( N_{\text{max}} \): Pascal’s \( d \)-simplex:

<table>
<thead>
<tr>
<th>( N_{\text{max}} = 0 )</th>
<th>( N_{\text{max}} = 1 )</th>
<th>( N_{\text{max}} = 2 )</th>
<th>( N_{\text{max}} = 3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( {0,0,0,0} )</td>
<td>( {0,1,0,0} )</td>
<td>( {0,0,1,0} )</td>
<td>( {0,2,0,0} )</td>
</tr>
<tr>
<td></td>
<td>( {1,0,0,0} )</td>
<td>( {0,1,0,0} )</td>
<td>( {2,0,0,0} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( {0,0,2,0} )</td>
<td>( {2,0,1,0} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( {0,1,1,0} )</td>
<td>( {2,0,0,1} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( {0,1,0,1} )</td>
<td>( {2,0,1,1} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( {1,0,0,1} )</td>
<td>( {2,0,0,1} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( {1,1,0,0} )</td>
<td>( {2,1,0,0} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( {1,1,0,1} )</td>
<td>( {2,1,0,1} )</td>
</tr>
</tbody>
</table>

Each sphere represents a \( d \times d \) complex-valued matrix

---

\(^1\) Strümpfer & Schulten: Open Quantum Dynamics Calculations with the Hierarchy Equations of Motion on Parallel Computers, J Chem Theory Comput, 2012, 8, 2808
Solving the hierarchy (HEOM) on the GPU. Program flow

**GPU** allocate array of $10^6 \times d \times d$ matrices indexed by Pascal's d-simplices up to fixed $N_{\text{max}}$ truncation (5 GB memory)

**CPU** compute connectivity graph,
transfer list of connected memory addresses to GPU
(0.5 GB memory)

— begin iteration step

**GPU** compute time-derivative $\frac{d}{dt} \sigma^{\bar{n}}$:

- kernel 1 constant matrix $\times$ vector of matrices commutator
- kernel 2 matrix-multiply-add with connected matrices $\sigma^{\bar{n}} \pm$

**GPU** Runge-Kutta integration to obtain $\sigma^{\bar{n}}(t + \Delta t)$

- kernel 3 vector-add on all matrix elements

**CPU** every $N_{\text{steps}}$ fetch $\rho(t + \Delta t) \equiv \sigma^{\bar{0}}(t + \Delta t)$

— repeat step 100,000 times

connectivity of matrices addressed by d-tuples

Results: compare with experiment

Theory

Experiment (various temperatures)

computed with GPU-HEOM

✓ model parameters directly taken from atomistic models, no fitting
✓ GPU-HEOM only available approach,
  computation-time ca 4 GPU-days [NVIDIA K20c]
  reasonable good agreement with experiment,
  supports quantum-mechanical coherence

---

2 Panitchayangkoon G et al. PNAS, 2010, 107, 12766
Cloud computing for reproducible results

Only GPU algorithm (GPU-HEOM)\(^1\) allows one to study FMO complex.

**problem** not all researcher have GPUs+code available

**solution**

- team-up with nanoHUB NSF-supported network for computational nanotechnology\(^2\)
  - more than 2000 program runs within one year, reference tool and cited in publications
  - location of GPU-HEOM users, 1st nanoHUB GPU-tool

---


\(^2\)Kreisbeck & TK: Exciton dynamics lab for light-harvesting complexes (GPU-HEOM) @ nanoHUB
The nanoHUB platform

- user interface runs over VNC server; sessions (re)opened from web-browser
- compute back-end: GPU M2090 cluster at Purdue University
- users can download data, share sessions
- great resource for collaborations and providing supporting material for publications

---

1 Madhavan, Zentner & Klimeck, *Learning and research in the cloud*, Nature Nanotech 2013, 8, 786
contains most abundant photosynthetic system on earth
light-harvesting complex regulates efficiency (clouds!)
complex system with $3 \times 14$ coupled chlorophylls (compared to 7 for FMO)
laser spectroscopy data available
computationally challenging, large memory required
Porting GPU-HEOM to OpenCL

**goal:** run on large-memory devices without rewriting all GPU code

- ✓ code relatively easy to transfer from CUDA to OpenCL
- ✓ OpenCL runs as fast as CUDA on GPU
  - OpenCL kernel for many-core CPU: less threads with more work/thread
  - OpenCL kernel not optimized for Intel MIC (requires data reorganization)
- 😊 OpenCL on CPU allocates up to 128 GB memory for larger systems
OpenCL: targeting CPU and GPU

OpenCL [CPU]

1 workdim=1;
2 // worksize equal # matrices
3 // no fast shared memory
4
globalWorkSize[] = { N_matrices };
5 clEnqueueNDRangeKernel(...,workdim,0,
6 globalWorkSize,NULL,0,0,&event);

OpenCL [GPU]

1 workdim=2;
2 // more threads: # matrices × # elements
3 // use shared (local) memory
4 localWorkSize[] = {16,16};
5 ngridy=16;
6 ngridx=N_matrices*N_elements/(ngridy*16*16);
7 localWorkSize[] = {16,16}
8 globalWorkSize[] = {ngridx*16,ngridy*16};
9 clEnqueueNDRangeKernel(...,workdim,0,
10 globalWorkSize,NULL,0,0,&event);

▶ GPUs perform best with blocks of threads, naturally organized in 2d-grids
▶ CPUs perform best with less threads, but doing more work per thread: matrix-matrix multiplication with three loops inside the kernel
▶ Intel MIC is somewhere in between, requires to change data-layout
Writing portable code: from Intel MIC to the smartphone GPU

- getting the OpenCL code running on varying platforms helps to iron out errors and to write portable code
- be prepared for future developments and for running on many-core CPU systems

😊 problem: often poor vendor support for OpenCL

OpenCL-HEOM on Nexus 4 smartphone

see our blog http://quantumdynamics.wordpress.com
Future work

- OpenCL offers good possibilities to run on both, GPUs and CPUs
- Kernels run even without modifications, high performance requires to adjust memory access, # threads
- GPUs are so-far the best “dense” number-crunchers, only limited by memory size (K40 helps)
- Needed: better OpenCL compilers, algorithms for heterogenous hardware (mixture of GPU/many core CPU/Intel MIC)
Acknowledgements

Dr. C. KREISBECK
Harvard University

Dr. B. HEIN
Regensburg, Germany

Dr. M. RODRÍGUEZ
Barcelona, Spain

Prof. A. ASPURU-GUZIK
Harvard University

supported by:

- DFG (German Research Foundation) 2007-2014
- Harvard University, Department of Physics, Chemistry, and School of Applied Sciences and Engineering
- DAAD Acciones Integradas Hispano-Alemanas
- Universitäts-Stiftung Vielberth Regensburg

http://www.quantumdynamics.de