Molecular Dynamics

Alexander B. Pacheco

User Services Consultant
LSU HPC & LONI
sys-help@loni.org

LSU HPC Training Series
Louisiana State University
February 8, 2011

Outline

1. Introduction
2. Fundamentals of Molecular Dynamics
3. *Ab Initio* Molecular Dynamics Theory
4. Computational Chemistry Programs
    - Gaussian
    - GAMESS
    - CPMD
    - Amber
    - LAMMPS
    - NAMD
    - Gromacs
5. Benchmarks
6. Molecular Dynamics on GPU

What is Computational Chemistry

- **Computational Chemistry** is a branch of chemistry that uses principles of computer science to assist in solving chemical problems.
- Uses the results of theoretical chemistry, incorporated into efficient computer programs.
- Application to single molecule, groups of molecules, liquids or solids.
- Calculates the structure and properties such as relative energies, charge distributions, dipole and multipole moments, spectroscopy, reactivity, etc.
- Computational Chemistry Methods range from
  1. Highly accurate (*Ab-initio*,DFT) feasible for small systems
  2. Less accurate (semi-empirical)
  3. Very Approximate (Molecular Mechanics, Classical Mechanics) large systems

Theoretical Chemistry: broadly can be divided into two main categories

1. Static Methods ⟹ Time-Independent Schrödinger Equation
    - Quantum Chemical/*Ab Initio* /Electronic Structure Methods
    - Molecular Mechanics
2. Dynamical Methods ⟹ Time-Dependent Schrödinger Equation
    - Classical Molecular Dynamics
    - Semi-classical and *Ab-Initio* Molecular Dynamics

Outline

Why Molecular Dynamics?

- Electronic Structure Methods are applicable to systems in gas phase under low pressure (vaccum).
- Majority of chemical reactions take place in solution at some temperature with biological reactions usually at specific pH's.
- Calculating molecular properties taking into account such environmental effects which can be dynamical in nature are not adequately described by electronic structure methods.

Molecular Dynamics

- Generate a series of time-correlated points in phase-space (a trajectory).
- Propagate the initial conditions, position and velocities in accordance with Newtonian Mechanics. $\mathbf{F} = m\mathbf{a} = -\boldsymbol{\nabla} V$
- Fundamental Basis is the **Ergodic Hypothesis**: the average obtained by following a small number of particles over a long time is equivalent to averaging over a large number of particles for a short time.

Applications of Molecular Dynamics

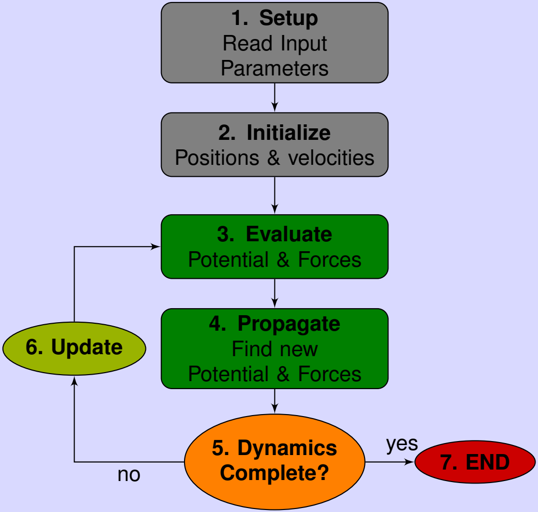
- Liquids, transport phenomena like viscosity and heat flow
- Crystal structure, defects
- Solids: Fracture, Friction between two solids
- Surface Phenomena, adsorption, diffusion
- Clusters are a bridge between molecular systems and solids, catalysis
- Biomolecules
- Electronic Properties and Dynamics

Limitations of Molecular Dynamics

- Use of classical forces: Nuclear quantum effects become important for lighter nuclei (e.g. H, He), low temperatures.
- How realistic are the forces?
- How long should the simulation run?
- Size of system?

General Schematic for MD Program

1. Setup: Read input parameters
2. Initialize: Obtain initial positions and velocities
3. Evaluate: Potential Energy and Forces on nuclei
4. Propagate nuclei using an appropriate time integration algorithm
5. Check if Dynamics is complete.
6. If incomplete update variables and goto Step 3.
7. If complete end dynamics or carry out all required analysis.



Models for Physical System

- Model for the physical system being simulated
- Choose the potential which is a function $\mathcal{V}(\mathbf{r}_1 \cdots \mathbf{r}_N)$ of the positions of the nuclei, representing the potential energy of the system when the atoms are arranged in that specific configuration.
- Forces are obtained as gradients of the potential

$$\mathbf{F}_i = -\nabla_{\mathbf{r}_i} \mathcal{V}(\mathbf{r}_1 \cdots \mathbf{r}_i \cdots \mathbf{r}_N)$$

- Simplest choice for potential is a sum of pairwise interactions

$$\mathcal{V}(\mathbf{r}_1 \cdots \mathbf{r}_N) = \sum_i \sum_{j>i} V(|\mathbf{r}_i - \mathbf{r}_j|)$$

Potential Energy Functions

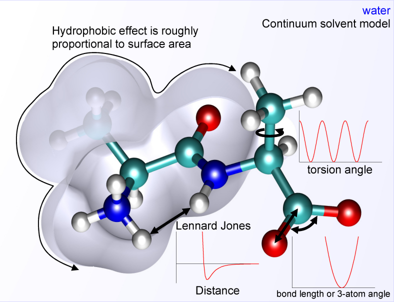
- Pair Potentials:
  1. Attraction
     - Long range
     - Dispersive forces, instantaneous dipole interactions
  2. Repulsion
     - Short range
     - Exchange forces, overlap of electron cloud.
- Lennard-Jones (LJ) Potential

$$U(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right]$$

  1. The LJ potential decays rapidly, significant computation time can be saved by neglecting pair interactions beyond a cut-off. [for $r_c > 2.5\sigma,\ U(r) = 0$]
- Potential from Molecular Mechanics Force fields like AMBER, CHARMM, Drieding etc or from electronic structure calculations.

General form of Molecular Mechanics equations

$$
\begin{aligned}
E &= E_{\text{bond}} + E_{\text{angle}} + E_{\text{torsion}} + E_{\text{vdW}} + E_{\text{elec}} \\
&= \frac{1}{2}\sum_{\text{bonds}} K_b(b-b_0)^2 && \text{Bond} \\
&+ \frac{1}{2}\sum_{\text{angles}} K_\theta(\theta-\theta_0)^2 && \text{Angle} \\
&+ \frac{1}{2}\sum_{\text{dihedrals}} K_\phi\left[1+\cos(n\phi)\right]^2 && \text{Torsion} \\
&+ \sum_{\text{nonbonds}} \left\{ \begin{array}{l} \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \\ + \frac{q_1 q_2}{Dr} \end{array} \right. && \begin{array}{l}\text{van der Waals} \\ \text{Electrostatics}\end{array}
\end{aligned}
$$



Picture taken from
http://en.wikipedia.org/wiki/Molecular_mechanics

Initial Positions

1. From experimental results
2. Assign to lattice positions

Initial Velocity

- Trajectories from previous simulation
- Set to desired temperature of simulation
    1. Assign a Maxwell -Boltzmann distribution

$$f_v(v_x) = \sqrt{\frac{m}{2\pi kT}} \exp\left[\frac{-mv_x^2}{2kT}\right]$$

    2. Random Uniform distribution: Random velocities scaled to desired temperature

$$v_{desired} = \sqrt{\frac{T_{desired}}{T_{current}}} v_{current}$$

- Total linear momentum is zero

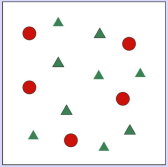
$$\sum_i m_i v_i = 0 \qquad \text{or} \qquad v_j^{desired} = v_j^{current} - \frac{\sum_i m_i v_i^{current}}{m_j}$$

Boundary Conditions

- What do we do at the boundaries of the simulated system?
    1. Nothing: Not particularly useful for infinite systems but good for single isolated or finite systems.
    2. Use Periodic Boundary Conditions (PBC) for simulating liquids, solids and clusters.

Periodic Boundary Conditions

- Consider a small subset or representative size of the infinite or large system and replicate to infinity (or your desired system size).



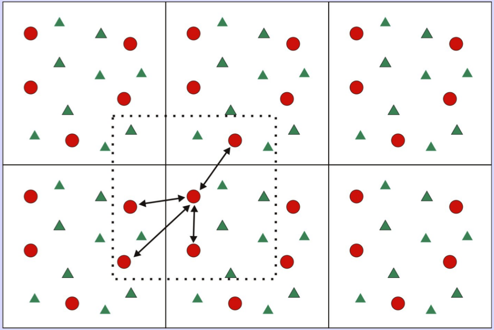
- For each particle in our simulation box, there are infinite particles located at positions

  $$\mathbf{r} + l\mathbf{a} + m\mathbf{b} + n\mathbf{c}\,,\ (l, m, n = -\infty \cdots \infty)$$

  where **a**, **b** and **c** are the lattice dimensions of the box.
- Thus each particle interacts not only with the other particles in the simulation box but also with their images in every other box.
- The simulated box and its images have the same size, shape and number of atoms with the same position, velocity and acceleration.
- How to calculate these infinite interactions which will undoubtedly increase the simulation cost?

Minimum Image Criterion

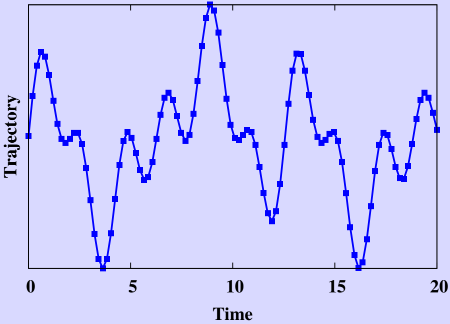
- If the potential has a finite range, say R_c, then two particles at a distance greater than R_c do not interact with each other.
- If the size of the simulation box is large than $2R_c$, then each particle i will at most interact with one particle j in the simulation box or the neighboring box.
- Thus interaction between particle i with particle j and its images in the replicated boxes will only consist of one interaction between the closest particle.



- Periodic Boundary Conditions are well suited for modeling infinite systems such as liquids and solids.
- For surface simulation, a model of a slab is used i.e. a thick slice of the material delimited with two free surfaces.
- This is done by removing the PBC from one direction, say z direction while maintaining PBC in the xy plane.
- Thus the system is infinite in the xy plane and finite in the z direction.
- Removing PBC in two directions gives rise to a wire model.
- Removing PBC's completely gives rise to a cluster model.

Time Integration

- The main ingredient of Molecular Dynamics is the time integration algorithm: integrate equation of motions of particles to follow trajectories.
- Based on finite difference methods: time is discretized onto a finite grid with the time step Δt being the distance between consecutive points on the grid.



- From position and time derivative at time t, obtain positions and time derivatives at time $t + \Delta t$

Verlet Algorithm

- Taylor expansion around $\mathbf{r}(t)$

$$\begin{aligned}
\mathbf{r}(t+\Delta t) &= 2\mathbf{r}(t) + \mathbf{r}(t-\Delta t) + \mathbf{a}(t)(\Delta t)^2 + \mathcal{O}(\Delta t)^4 \\
\mathbf{a}(t) &= (-1/m)\nabla\mathcal{V}(\mathbf{r}(t)) \\
\mathbf{v}(t) &= \frac{\mathbf{r}(t+\Delta t) - \mathbf{r}(t-\Delta t)}{2\Delta t}
\end{aligned}$$

Velocity Verlet Algorithm

- Obtain velocity at half step and position at full step.

$$\begin{aligned}
\mathbf{v}(t+\Delta t/2) &= \mathbf{v}(t) + 1/2\mathbf{a}(t)\Delta t \\
\mathbf{r}(t+\Delta t) &= \mathbf{r}(t) + \mathbf{v}(t+\Delta t/2)\Delta t \\
\mathbf{a}(t+\Delta t) &= (-1/m)\nabla\mathcal{V}(\mathbf{r}(t+\Delta t)) \\
\mathbf{v}(t+\Delta t) &= \mathbf{v}(t+\Delta t/2) + 1/2\mathbf{a}(t+\Delta t)\Delta t
\end{aligned}$$

Predictor Corrector Algorithm

- Predictor Step: Predict position and time derivatives time $t + \Delta t$ by Taylor expansion at time t.
- Compute Forces as gradient of potentials at the predicted positions. The difference between the predicted acceleration and the calculated acceleration is the "error signal"
$$\Delta a(t + \Delta t) = a^C(t + \Delta t) - a^P(t + \Delta t)$$
- Corrector Step: Use the "error signal" to correct the positions and its time derivatives.
$$\begin{aligned} r^C(t + \Delta t) &= r^P(t + \Delta t) + c_0 \Delta a(t + \Delta t) \\ v^C(t + \Delta t) &= v^P(t + \Delta t) + c_1 \Delta a(t + \Delta t) \\ a^C(t + \Delta t) &= a^P(t + \Delta t) + c_2 \Delta a(t + \Delta t) \\ b^C(t + \Delta t) &= b^P(t + \Delta t) + c_3 \Delta a(t + \Delta t) \end{aligned}$$
- The coefficients maximizes stability and are dependent on specific algorithm
- **Gear Predictor Corrector Algorithm:**
$c_0 = 1/6$, $c_1 = 5/6$, $c_2 = 1$ and $c_3 = 1/3$.

Analysis of Trajectory I

- Kinetic Energy

$$K = \frac{1}{2}\sum_i m_i v_i^2$$

- Temperature: from average kinetic energy using equipartition theorem

$$T = \frac{2K}{3k_B}$$

- Pressure: from virial theorem

$$P = \frac{Nk_BT}{V} - \frac{1}{N_{dim}}\left\langle \sum_i^N \mathbf{r}_i \cdot \mathbf{F}_i \right\rangle$$

  where N is number of particles and N_{dim} is dimensionality of the system.

- Diffusion Coefficient: related to mean square displacement

$$D = \frac{1}{2N_{dim}} \lim_{t\to\infty} \frac{\langle |\mathbf{r}(t) - \mathbf{r}(0)|^2 \rangle}{t}$$

*Analysis of Trajectory **II***

- Spectral analysis
  1. FT-VAC: Fourier Transform of Velocity Auto-Correlation function

     $$V(\omega) = \frac{1}{2\pi}\int \exp(-\imath\omega t)\,\langle \mathbf{v}(t)\cdot\mathbf{v}(0)\rangle$$

     if mass weighted velocities $\boldsymbol{\mathcal{V}}_i = \sqrt{m_i}\mathbf{v}_i$ are used, then $V(\omega)$ is the kinetic energy spectra.
  2. FT-DAC: Fourier Transform of Dipole Auto-Correlation function, related to IR spectra

     $$S(\omega) = \frac{1}{2\pi}\int \exp(-\imath\omega t)\,\langle \boldsymbol{\mu}(t)\cdot\boldsymbol{\mu}(0)\rangle$$

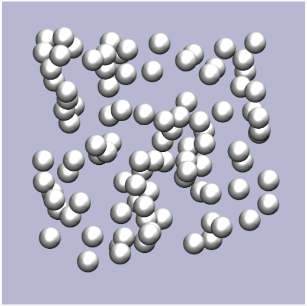
  3. Short-Time Fourier Transform (STFT): 2D time-frequency spectra to simulate pump-probe experiments

     $$\mathcal{K}(t,\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} dt'\,\left\langle \boldsymbol{\mathcal{V}}(t)\cdot\boldsymbol{\mathcal{V}}(t')\right\rangle H(t,t')\exp(-\imath\omega t')$$

     where $H(t,t')$ is a window function

Example code for MD

- On LONI and LSU HPC Linux systems: /home/apacheco/CompChem/MD_Prog2Prod
- Equilibration of liquid Hydrogen.
- Courtesy: Matt McKenzie, formerly LSU HPC now at NICS.
- Input File: fort.40
- Output File: fort.44 (energy data) and fort.77 (xyz file of dynamics)
- In directory crystal, crystal.f90 to generate lattice structure, courtesy Furio Ercolessi `http://www.fisica.uniud.it/~ercolessi/md`



1. Introduction
2. Fundamentals of Molecular Dynamics
3. *Ab Initio* Molecular Dynamics Theory
4. Computational Chemistry Programs
    - Gaussian
    - GAMESS
    - CPMD
    - Amber
    - LAMMPS
    - NAMD
    - Gromacs
5. Benchmarks
6. Molecular Dynamics on GPU

Ab Initio Molecular Dynamics: Theory

- Solve the time-dependent Schrödinger equation

$$\imath\hbar\frac{\partial}{\partial t}\Psi(\mathbf{R},\mathbf{r},t) = \hat{H}\Psi(\mathbf{R},\mathbf{r},t)$$

with

$$\Psi(\mathbf{R},\mathbf{r},t) = \chi(\mathbf{R},t)\Phi(\mathbf{r},t)$$

and

$$\hat{H} = -\sum_I \frac{\hbar^2}{2M_I}\nabla_I^2 + \underbrace{\frac{-\hbar^2}{2m_e}\nabla_i^2 + V_{n-e}(\mathbf{r},\mathbf{R})}_{H_e(\mathbf{r},\mathbf{R})}$$

- Obtain coupled equations of motion for electrons and nuclei: Time-Dependent Self-Consistent Field (TD-SCF) approach.

$$\imath\hbar\frac{\partial\Phi}{\partial t} = \left[-\sum_i \frac{\hbar^2}{2m_e}\nabla_i^2 + \langle\chi|V_{n-e}|\chi\rangle\right]\Phi$$

$$\imath\hbar\frac{\partial\chi}{\partial t} = \left[-\sum_I \frac{\hbar^2}{2M_I}\nabla_I^2 + \langle\Phi|H_e|\Phi\rangle\right]\chi$$

- Define nuclear wavefunction as

$$\chi(\mathbf{R},t) = A(\mathbf{R},t)\exp\left[\imath S(\mathbf{R},t)/\hbar\right]$$

where A and S are real.

- Solve the time-dependent equation for nuclear wavefunction and take classical limit ($\hbar \to 0$) to obtain

$$\frac{\partial S}{\partial t} + \sum_I \frac{\hbar^2}{2M_I}(\nabla_I S)^2 + \langle\Phi|H_e|\Phi\rangle = 0$$

an equation that is isomorphic with the Hamilton-Jacobi equation with the classical Hamilton function given by

$$\mathcal{H}(\{\mathbf{R}_I\},\{\mathbf{P}_I\}) = \sum_I \frac{\hbar^2}{2M_I}\mathbf{P}_I^2 + V(\{\mathbf{R}_I\})$$

where

$$\mathbf{P}_I \equiv \nabla_I S \qquad \text{and} \qquad V(\{\mathbf{R}_I\}) = \langle\Phi|H_e|\Phi\rangle$$

- Obtain equations of nuclear motion from Hamilton's equation

$$\frac{d\mathbf{P}_I}{dt} = -\frac{d\mathcal{H}}{d\mathbf{R}_I} \Rightarrow M\ddot{\mathbf{R}}_I = -\nabla_I V$$

$$\frac{d\mathbf{R}_I}{dt} = \frac{d\mathcal{H}}{d\mathbf{P}_I}$$

- Replace nuclear wavefunction by delta functions centered on nuclear position to obtain

$$\imath\hbar\frac{\partial\Phi}{\partial t} = H_e(\mathbf{r},\{\mathbf{R}_I\})\Phi(\mathbf{r};\{\mathbf{R}_I\},t)$$

- This approach of simultaneously solving the electronic and nuclear degrees of freedom by incorporating feedback in both directions is known as **Ehrenfest Molecular Dynamics**.
- Expand Φ in terms of many electron wavefunctions or determinants

$$\Phi(\mathbf{r};\{\mathbf{R}_I\},t) = \sum_i c_i(t)\Phi_i(\mathbf{r};\{\mathbf{R}_I\})$$

  with matrix elements

$$H_{ij} = \langle\Phi_i|H_e|\Phi_j\rangle$$

- Inserting Φ in the TDSE above, we get

$$\imath\hbar\dot{c}_i(t) = c_i(t)H_{ii} - \imath\hbar\sum_{I,i}\dot{\mathbf{R}}_I\mathbf{d}_I^{ij}$$

  with non-adiabatic coupling elements given by

$$\mathbf{d}_I^{ij}(\mathbf{R}_I) = \langle\Phi_i|\nabla_I|\Phi_j\rangle$$

- Up to this point, no restriction on the nature of Φ_i i.e. adiabatic or diabatic basis has been made.

- Ehrenfest method rigorously includes non-adiabatic transitions between electronic states within the framework of classical nuclear motion and mean field (TD-SCF) approximation to the electronic structure.
- Now suppose, we define $\{\Phi_i\}$ to be the adiabatic basis obtained from solving the time-independent Schrödinger equation,

$$H_e(\mathbf{r}, \{\mathbf{R}_I\})\Phi_i(\mathbf{r}; \{\mathbf{R}_I\}) = E_i(\{\mathbf{R}_I\})\Phi_i(\mathbf{r}; \{\mathbf{R}_I\})$$

- The classical nuclei now move along the adiabatic or Born-Oppenheimer potential surface. Such dynamics are commonly known as **Born-Oppenheimer Molecular Dynamics** or BOMD.
- If we restrict the dynamics to only the ground electronic state, then we obtain ground state BOMD.
- If the Ehrenfest potential $V(\{\mathbf{R}_I\})$ is approximated to a global potential surface in terms of many-body contributions $\{v_n\}$.

$$V(\{\mathbf{R}_I\}) \approx V_e^{approx}(\mathbf{R}) = \sum_{I=1}^{N} v_1(\mathbf{R}_I) + \sum_{I>J}^{N} v_2(\mathbf{R}_I, \mathbf{R}_J) + \sum_{I>J>K}^{N} v_3(\mathbf{R}_I, \mathbf{R}_J, \mathbf{R}_K) + \cdots$$

- Thus the problem is reduced to purely classical mechanics once the $\{v_n\}$ are determined usually Molecular Mechanics Force Fields. This class of dynamics is most commonly known as **Classical Molecular Dynamics**.

- Another approach to obtain equations of motion for ab-initio molecular dynamics is to apply the Born-Oppenheimer approximation to the full wavefunction $\Psi(\mathbf{r}, \mathbf{R}, t)$

$$\Psi(\mathbf{r}, \mathbf{R}, t) = \sum_k \chi_k(\mathbf{R}, t)\Phi_k(\mathbf{r}; \mathbf{R}(t))$$

where

$$H_e\Phi_k(\mathbf{r}; \mathbf{R}(t)) = E_k(\mathbf{R}(t))\Phi_k(\mathbf{r}; \mathbf{R}(t))$$

- Assuming that the nuclear dynamics does not change the electronic state, we arrive at the equation of motion for nuclear wavefunction

$$\imath\hbar\frac{\partial}{\partial t}\chi_k(\mathbf{R}, t) = \left[\sum_I -\frac{\hbar^2}{2M_I}\nabla_I^2 + E_k(\mathbf{R})\right]\chi_k(\mathbf{R}, t)$$

- The Lagrangian for this system is given by.

$$\mathcal{L} = \hat{T} - \hat{V}$$

- Corresponding Newton's equation of motion are then obtained from the associated Euler-Lagrange equations,

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{R}}_I} = \frac{\partial \mathcal{L}}{\partial \mathbf{R}_I}$$

- The Lagrangian for ground state BOMD is

$$\mathcal{L}_{\text{BOMD}} = \sum_I \frac{1}{2} M_I \dot{\mathbf{R}}_I^2 - \min_{\Phi_0} \langle \Phi | H_e | \Phi \rangle$$

and equations of motions

$$M_I \ddot{\mathbf{R}}_I = \frac{d}{dt}\frac{\partial \mathcal{L}_{\text{BOMD}}}{\partial \dot{\mathbf{R}}_I} = \frac{\partial \mathcal{L}_{\text{BOMD}}}{\partial \mathbf{R}_I} = -\nabla_I \min_{\Phi_0} \langle \Phi | H_e | \Phi \rangle$$

Extended Lagrangian Molecular Dynamics (ELMD)

Extend the Lagrangian by adding kinetic energy of fictitious particles and obtain their equation of motions from Euler-Lagrange equations.

Molecular Orbitals: $\{\phi_i\}$

Density Matrix: $P_{\mu\nu} = \sum_i c^*_{\mu i} c_{\nu i}$

Car-Parrinello Molecular Dynamics (CPMD) : CPMD & NWChem

$$\mathcal{L}_{\text{CPMD}} = \sum_I \frac{1}{2} M_I \dot{\mathbf{R}}_I^2 + \sum_i \frac{1}{2} \mu_i \langle \dot{\phi}_i | \dot{\phi}_i \rangle - \langle \Phi_0 | H_e | \Phi_0 \rangle + \text{constraints}$$

R. Car and M. Parrinello, Phys. Rev. Lett. 55 (22), 2471 (1985)

Atom centered Density Matrix Propagation (ADMP) : Gaussian 03/09

$$\mathcal{L}_{\text{ADMP}} = \frac{1}{2} \mathrm{Tr}(\mathbf{V}^T \mathbf{M} \mathbf{V}) + \frac{1}{2} \mu \mathrm{Tr}(\dot{\mathbf{P}} \dot{\mathbf{P}}) - E(\mathbf{R}, \mathbf{P}) - \mathrm{Tr}[\mathbf{\Lambda}(\mathbf{PP} - \mathbf{P})]$$

H. B. Schlegel, J. M. Millam, S. S. Iyengar, G. A. Voth, A. D. Daniels, G. E. Scuseria, M. J. Frisch, J. Chem. Phys. 114, 9758 (2001)

curvy-steps ELMD (csELMD) : Q-Chem

$$\mathcal{L}_{\text{csELMD}} = \sum_I \frac{1}{2} M_I \dot{\mathbf{R}}_I^2 + \frac{1}{2} \mu \sum_{i<j} \dot{\Delta}_{ij} - E(\mathbf{R}, \mathbf{P}); \quad \mathbf{P}(\lambda) = e^{\lambda \Delta} \mathbf{P}(0) e^{-\lambda \Delta}$$

J.M. Herbert and M. Head-Gordon, J. Chem. Phys. 121, 11542 (2004)

- Electronic energy obtained from
  - Molecular Mechanics ⇒ Classical Molecular Dynamics
    1. LAMMPS
    2. NAMD
    3. Amber
    4. Gromacs
  - Ab-Initio Methods ⇒ Quantum or Ab-Initio Molecular Dynamics
    1. Born-Oppenheimer Molecular Dynamics: Gaussian, GAMESS
    2. Extended Lagrangian Molecular Dynamics: CPMD, Gaussian (ADMP), NWCHEM(CPMD), VASP, QChem (curvy-steps ELMD)
    3. Time Dependent Hartree-Fock and Time Dependent Density Functional Theory: Gaussian, GAMESS, NWCHEM, QChem
    4. Multiconfiguration Time Dependent Hartree(-Fock), MCTDH(F)
    5. Non-Adiabatic and Ehrenfest Molecular Dynamics, Multiple Spawning, Trajectory Surface Hopping
    6. Quantum Nuclei: QWAIMD(Gaussian), NEO(GAMESS)

Classical Molecular Dynamics

- Advantages
  1. Large Biological Systems
  2. Long time dynamics
- Disadvantages
  1. Cannot describe Quantum Nuclear Effects

Ab Initio and Quantum Dynamics

- Advantages
  1. Quantum Nuclear Effects
- Disadvantages
  1. $\sim$ 100 atoms
  2. Full Quantum Dynamics ie treating nuclei quantum mechanically: less than 10 atoms
  3. Picosecond dynamics at best

Outline

1. Introduction
2. Fundamentals of Molecular Dynamics
3. *Ab Initio* Molecular Dynamics Theory
4. Computational Chemistry Programs
    - Gaussian
    - GAMESS
    - CPMD
    - Amber
    - LAMMPS
    - NAMD
    - Gromacs
5. Benchmarks
6. Molecular Dynamics on GPU

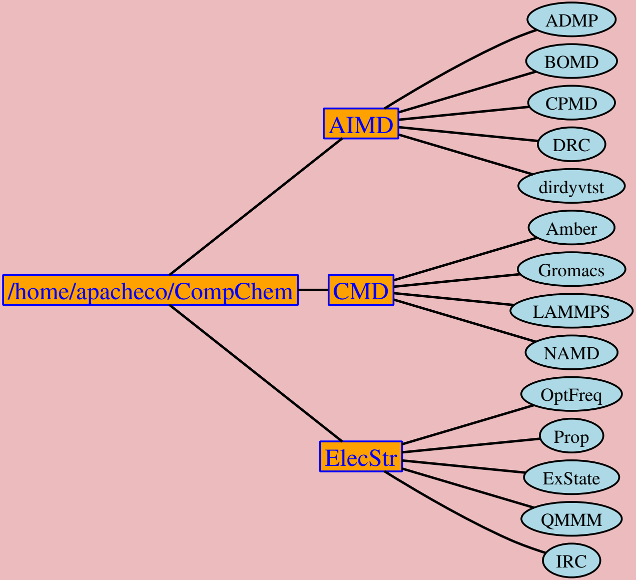
Software Installed on LONI & LSU HPC Systems

Software	QB	Eric	Louie	Oliver	Painter	Poseidon	Bluedawg
Amber	✓	✓	✓	✓	✓	✓	
Gromacs	✓	✓	✓	✓	✓	✓	✓
LAMMPS	✓	✓	✓	✓	✓	✓	✓
NAMD	✓	✓	✓	✓	✓	✓	✓
CPMD	✓	✓	✓	✓	✓	✓	✓
GAMESS	✓	✓	✓	✓	✓	✓	
Gaussian		✓	✓	✓	✓		✓
NWChem	✓	✓	✓	✓	✓	✓	✓

Software	Tezpur	Philip	Pelican	Pandora
Amber	✓	✓	✓	✓
Gromacs	✓	✓	✓	✓
LAMMPS	✓	✓	✓	✓
NAMD	✓	✓	✓	✓
CPMD	✓		✓	✓
GAMESS	✓	✓		✓
Gaussian	✓	✓	✓	✓
NWChem	✓		✓	✓

Computational Chemistry Programs

- Other Software: Q-Chem, SIESTA ,CHARMM, VASP, Quantum Expresso, Octopus
- `http://en.wikipedia.org/wiki/Quantum_chemistry_computer_programs`
- `http://www.ccl.net/chemistry/links/software/index.shtml`
- `http://www.redbrick.dcu.ie/~noel/linux4chemistry/`


/home/apacheco/CompChem
AIMD
ADMP
BOMD
CPMD
DRC
dirdyvtst
CMD
Amber
Gromacs
LAMMPS
NAMD
ElecStr
OptFreq
Prop
ExState
QMMM
IRC

Gaussian I

- Atom-centered Density Matrix Propagation (ADMP) is an extended Lagrangian technique similar to CPMD which propagates the density matrix constructed from atom-centered gaussian functions.
- ADMP is part of Gaussian 03/09 (link 121) and can carry out ADMP as well as BOMD calculations for ground state electronic structure.
- To run BOMD calculations via link 121, add the keyword "fullscf" alongwith other admp keywords

```
%chk=water-dimer-admp.chk
%mem=512mb
%nprocshared=4

#n b3lyp/6-31+G(D) admp(stepsize=1000,maxpoints=10000,NKE=7125)

ADMP: 1ps dynamics of water dimer at 300K with dt=0.1fs and mu=0.1amu
  Initial geometry is taken from CPMD calculation on water-dimer
NKE => initial energy in microH = Temp*(1.5*(N-1)*k)/(au/J)*1e6

0 1
 O     -1.378379      0.000490      0.080874
 O      1.539977     -0.000171     -0.118836
 H     -1.809450     -0.765701     -0.319056
 H     -1.806537      0.764674     -0.319022
 H      0.579977     -0.001581      0.034614
 H      1.932548     -0.002433      0.760717
```

*Gaussian **II***

- Gaussian 03/09 also provides BOMD (link 118) for post Hartree-Fock Methods
- Link 118 is expensive compared to Link 121; for ground state BOMD, Link 121 is recommended.

```
%chk=formaldehydedecomp-cas-bomd.chk
%nprocshared=4

#p cas(4,4)/6-31g(d,p) test
BOMD=(gradientonly,maxpoint=10,readmwvel)

h2co->h2+co

0,1
 C,0,0.000996921,0.,-0.0351483754
 O,0,0.0583751372,0.,1.1334216404
 H,0,1.5618411491,0.,-0.649341231
 H,0,0.2651862552,0.,-1.0795362823

0
1.0d14 -2.0d14 3.0d14
4.0d14 5.0d14 -6.0d14
7.0d14 -8.0d14 9.0d14
-10.0d14 11.0d14 12.0d14
```

GAMESS I

- Classical Trajectory Calculations within GAMESS are carried out using the DRC (Dynamic Reaction Coordinate) keyword.
- Not to be confused with IRC or Intrinsic Reaction Coordinate which finds the minimum energy path between the transition state and the reactants or products.

GAMESS II

```
 $CONTRL SCFTYP=RHF RUNTYP=DRC DFTTYP=B3LYP $END
 $BASIS GBASIS=N31 NGAUSS=6 NDFUNC=1 NPFUNC=1 DIFFSP=.T. $END
 $DATA
Water Dimer 6-31+G(D,P) OPT FREQ
C1
 O   8.0    -1.379701      0.002449      0.081416
 O   8.0     1.541565     -0.000860     -0.119477
 H   1.0    -1.818137     -0.764389     -0.315570
 H   1.0    -1.803571      0.759242     -0.315426
 H   1.0     0.581912     -0.007904      0.030592
 H   1.0     1.932153     -0.012152      0.759183
 $END
------- initial temp is 300K as in admp/bomd calculations
------- initial velocity vector taken from
------- /home/apacheco/CompChem/AIMD/CPMD/water-dimer/bo/g09-gms-input.vels
 $DRC NPRTSM=1 NSTEP=2000 DELTAT=0.5 NMANAL=.TRUE. EKIN=0.6 HESS=MIN
      VEL(1)=-6.41080769596195e-03   9.40773981527263e-03  2.67984190974773e-03
              7.75619391569166e-03  -3.36528283010609e-03 -3.16149484132192e-03
             -4.20098248607962e-02   6.91120029668994e-03  1.73187657665007e-02
              1.34786953525422e-02  -2.45099229646348e-02  1.65469867766813e-02
              9.32936500423984e-03  -3.08495101664572e-02 -1.96113477755845e-02
             -1.85637460617558e-03  -4.74561717078300e-02 -7.09855104721334e-03
 $END
------- hessian at optimized geometry calculated in
------- /home/apacheco/CompChem/ElecStr/OptFreq/GAMESS/water-dimer
 $HESS
ENERGY IS     -152.8035727224 E(NUC) IS       36.5901317657
 1  1 2.24977313E-01-3.08584011E-03 2.15513545E-01-1.03405650E-02-1.00996425E-06
 1  2-2.57280378E-03-1.05752527E-01-1.36622480E-01-1.10058074E-01-1.05761642E-01
 ... full hessian at optimized geometry ...
18  3-9.41697417E-06 6.49613332E-06 1.06343044E-02-1.62844886E-05 1.92196927E-03
18  4 1.96981522E-01 6.84369935E-04 4.34020160E-01
 $END
 $HESS2
ENERGY IS     -152.8035727224 E(NUC) IS       36.5901317657
 1  1 2.24977313E-01-3.08584011E-03 2.15513545E-01-1.03405650E-02-1.00996425E-06
 1  2-2.57280378E-03-1.05752527E-01-1.36622480E-01-1.10058074E-01-1.05761642E-01
  ... full hessian at reference geometry ...
18  2-4.42611411E-01-7.77174012E-06 9.07081889E-06 6.44610016E-06-7.99116222E-06
18  3-9.41697417E-06 6.49613332E-06 1.06343044E-02-1.62844886E-05 1.92196927E-03
18  4 1.96981522E-01 6.84369935E-04 4.34020160E-01
 $END
```

CPMD

- Car Parrinello Molecular Dynamics
- Typical problems CPMD is used for
  - Liquid Structures
  - Polarization effects
  - Bond breaking/formations
    - Proton transfer
  - Dynamic/thermal properties (e.g. diffusion)
  - Metadynamics
  - QM/MM
  - Path Integrals
  - TDDFT

CPMD Input

- Divided into sections
- Only sections pertaining to your simulated model at hand need to be present.
  - Common sections: INFO, CPMD, SYSTEM, ATOMS, DFT
  - Special sections: PIMD, PATH, RESP, LINRES, TDDFT, PROP, HARDNESS, CLASSIC, BASIS, VDW, QMMM
- Sections are defined `&Section_Name` and followed by `&END`, for example,

```
&ATOMS
  (information)
&END
```

- Each section has their own keywords
- Lines that do not match known keywords are ignored
- KEYWORDS HAVE TO BE IN ALL CAPS
  - kEYWORDS starting with a lower case character are ignored
  - Useful feature to re-use the input file
- Order of keywords is arbitrary

CPMD Input

```
&INFO
 Water Dimer
 CPMD at 300K
&END

&CPMD
 MOLECULAR DYNAMICS CP
 RESTART WAVEFUNCTION COORDINATES LATEST
 TRAJECTORY XYZ
 ISOLATED MOLECULE
 TEMPERATURE
  300.0D0
 MAXSTEP
  10336
 TIMESTEP
  4.0
 EMASS
  400.0
&END

&SYSTEM
 ANGSTROM
 SYMMETRY
  SIMPLE CUBIC
 CELL
  16.00 1.0 1.0  0.0  0.0  0.0
 CUTOFF
  60.0
&END
```

CPMD Input (contd)

```
&DFT
 FUNCTIONAL B3LYP
 SCREENED EXCHANGE ERFC
   0.1
 GC-CUTOFF
 5.0e-6
&END

&ATOMS
*O_SPRIK_BLYP.psp KLEINMAN-BYLANDER RAGGIO=1.2
 LMAX=P
  2
  -1.372075    0.000000    0.103925
   1.506485    0.000001   -0.120903
*H_SPRIK_BLYP.psp KLEINMAN-BYLANDER RAGGIO=1.2
 LMAX=S
  4
  -1.765171   -0.771860   -0.323273
  -1.765171    0.771862   -0.323267
   0.544563    0.000000    0.027907
   1.910502   -0.000003    0.754459
&END
```

AMBER

What is AMBER?

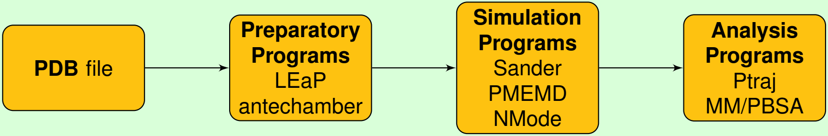
- AMBER, Assisted Model Building with Energy Refinement refers to two things
  1. A collective name for a suite of programs that allow users to carry out molecular dynamics simulation
  2. A set of molecular mechanical force fields for the simulation of biomolecules

Capabilities

- Classical MD (NVT,NPT,etc)
- Force Fields
- QM/MM
- Free Energy Calculations
- Structural and Trajectory analysis
- Parallelize dynamic codes
- Enhanced sampling (replica exchange MD)
- Explicit Solvent Models with PME
- Implicit Solvent Models with PB or GB approach

Basic Information Flow in AMBER

1. obtain and edit initial structure
2. prepare input parameter and topology file
3. run simulations and save trajectory
4. analyze output and trajectory files



PDB file

Preparatory Programs
LEaP
antechamber

Simulation Programs
Sander
PMEMD
NMode

Analysis Programs
Ptraj
MM/PBSA

Topology and Parameter File

Topology information includes

- atom types are assigned to identify different elements and different molecular orbital environments
- charges are assigned to each atom
- connectivities between atoms are established

Parameter information includes

- force constants necessary to describe the bond energy, angle energy, torsion energy, nonbonded interactions (van der Waals and electrostatics)
- other parameters for setting up the energy calculations (GB radii, FEP parameter sets)

LEaP

- Includes a tex-based interface - tleap and a graphical user interface - xleap
- Capabilities
    - Read AMBER force field information
    - Read and write files containing structural information (i.e.PDB files)
    - Construct new residues and molecules
    - Link together residues and create nonbonded complexes of molecules
    - Place counterions around a molecule; Solvate molecules; Mdoify internal coordinates within a molecule
    - Generate files that contain topology and parameters for AMBER
    - Set atomic charges, identify the position of disulphide bridges, delete bonds, addition of atoms, ions, etc ...

Using tleap/xleap

loading PDB file, adding the disulfide cross links and saving files

```
source leaprc.ff03
bpti = loadPdb 6pti.mod.pdb
bond bpti.5.SG bpti.55.SG
bond bpti.14.SG bpti.38.SG
bond bpti.30.SG bpti.51.SG
saveAmberParm bpti prmtop prmcrd
quit
```

Amber 11 Manual Page 17

Sander

- Sander: Simulated Annealing with NMR-Derived Energy Restraints
  - Energy minimization, molecular dynamics and NMR refinements
  - Free energy calculations (Umbrella Sampling; SMD; etc)
  - QM/MM implementation (EVB; semi-empirical/AMBER)
  - Polarizable force field (AMOEBA)
  - Enhanced Sampling (REMD; LES, etc)
- Usage: `sander -i mdin -o mdout -p prmtop -c inpcrd -r restrt`

mdin:	input control data for minimization/MD run.
mdout:	output file for user readable state info and diagnostics
prmtop:	molecular topology, force fields etc
inpcrd:	initial coordinates and velocities
restrt:	restart filename

Perform Minimization

200 steps of minimization, generalized Born solvent model

```
&cntrl
maxcycle=200,
imin=1,
cut=12.0,
igb=1,
ntb=0,
ntpr=10,
/
```

Amber 11 Manual Page 17

Run Sander

```
sander -i min.in -o 6pti.min1 -c prmcrd -r 6pti.min1.xyz
```

LAMMPS

- LAMMPS stands for Large-scale Atomic/Molecular Parallel Simulator.
- LAMMPS is a classical molecular dynamics code that models an ensemble of particles in a liquid, solid, or gaseous state designed to run efficiently on parallel computers.
- It can model atomic, polymeric, biological, metallic, granular, and coarse-grained systems using a variety of force fields and boundary conditions.
- LAMMPS can model systems with only a few particles up to millions or billions.
- LAMMPS is designed to be easy to modify or extend with new capabilities, such as new force fields, atom types, boundary conditions, or diagnostics.
- LAMMPS runs efficiently on single-processor desktop or laptop machines, but is designed for parallel computers.
- It is an open-source code, distributed freely under the terms of the GNU Public License (GPL).

Running LAMMPS

- LAMMPS doesn't
  1. Build molecular systems
  2. Assign force-dield coefficients auto-magically
  3. Compute lots of diagnostics on-the-fly
  4. Visualize your output
- LAMMPS version "4 May 2011" is installed on all LONI Dell Linux Clusters.
- Add the appropriate soft keys to your .soft file
  - +lammps-4May11-intel-11.1-mvapich-1.1
- Command line options for running LAMMPS
  - -in inputfile: specify input file
  - -log logfile: specify log file
  - -partition MxN L: Run on (MxN)+L processors with M partitions on N processors each and 1 partition with L processors
  - -screen file: Specify a file to write screen information

LAMMPS Input

- Reads an input script in ASCII format one line at a time.
- Input script consists of 4 parts
  1. Initialization: Set parameters that need to be defined before atoms are created or read-in from a file.
     `units, dimension, newton, processors, boundary, atom_style, atom_modify`
  2. Atom definition: `read_data, read_restart, lattice, region, create_box, create_atoms, replicate`
  3. Settings: Once atoms and molecular topology are defined, a variety of settings can be specified: force field coefficients, simulation parameters, output options, etc.
     `pair_coeff, bond_coeff, angle_coeff, dihedral_coeff, improper_coeff, kspace_style, dielectric, special_bonds, neighbor, neigh_modify, group, timestep, reset_timestep, run_style, min_style, min_modify, fix, compute, compute_modify, variable`
  4. Run a simulation: A MD is run using the run command. Energy minimization (molecular statics) is performed using the minimize command. A parallel tempering (replica-exchange) simulation can be run using the temper command.
- `http://lammps.sandia.gov/doc/Section_commands.html`

LAMMPS Input for 3d Lennard-Jones Melt

```
units lj                                     style of units used for a simulation
atom_style atomic                       what style of atoms to use in a simulatio

lattice fcc 0.8442                                          Lattice structure
region box block 0 20 0 20 0 20                    define simulation region
create_box 1 box                                        create simulation box
create_atoms 1 box                                   create atoms on lattice
mass 1 1.0
velocity all create 3.0 87287

pair_style lj/cut 2.5                         Formula for Pairwise Interaction
pair_coeff 1 1 1.0 1.0 2.5                  pairwise force field coefficients

neighbor 0.3 bin                                      pairwise neighbor lists
neigh_modify every 20 delay 0 check no

fix 1 all nve                                  constant NVE time integration

dump id all atom 10 dump.melt                               For visualization
thermo 50                                       output thermodynamics data
run 2500                                                     Simulation Steps
```

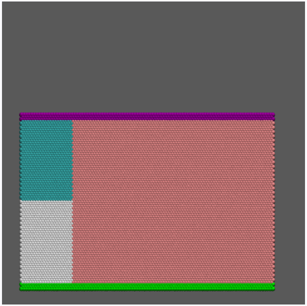
- "dump" command outputs snapshots of atom properties

  default format is simple : id, type, x, y, z
  other supported formats : XYZ, DCD, XTC

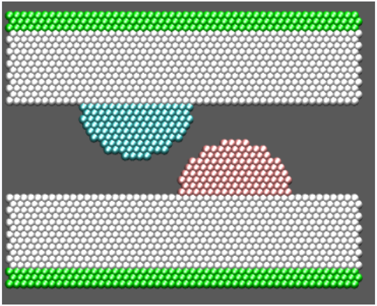
- Visualization using VMD

  `vmd -lammpstrj dumpfile`

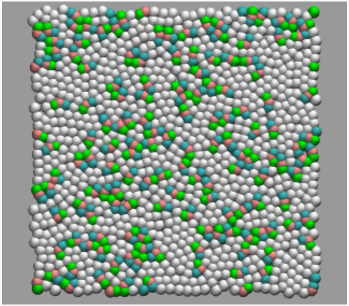
- The LAMMPS distribution includes an examples sub-directory with several sample problems.
- crack: crack propagation in a 2d solid



- The LAMMPS distribution includes an examples sub-directory with several sample problems.
- friction: frictional contact of spherical asperities between 2d surfaces



- The LAMMPS distribution includes an examples sub-directory with several sample problems.
- micelle: self-assembly of small lipid-like molecules into 2d bilayers



NAMD

- NAMD is a parallel molecular dynamics code designed for high-performance simulation of large biomolecular systems.
- Based on Charm++ parallel objects, NAMD scales to hundreds of processors on high-end parallel platforms and tens of processors on commodity clusters using gigabit ethernet.
- NAMD uses the popular molecular graphics program VMD for simulation setup and trajectory analysis, but is also file-compatible with AMBER, CHARMM, and X-PLOR.

What is needed to run NAMD?

- Protein Data Bank (pdb) file which stores atomic coordinates and/or velocities for the system.
- Protein Structure File (psf) which stores structural information of the protein, such as various types of bonding interactions.
- A force field parameter file. A force field is a mathematical expression of the potential which atoms in the system experience. CHARMM, X-PLOR, AMBER, and GROMACS are four types of force fields, and NAMD is able to use all of them. The parameter file defines bond strengths, equilibrium lengths, etc.
- A configuration file, in which the user specifies all the options that NAMD should adopt in running a simulation.

NAMD configuration file

- A NAMD configuration file contains a set of options and values which determine the exact behavior of NAMD, what features are active or inactive, how long the simulation should continue, etc.
- The following parameters are *required* for every NAMD simulation

  `numsteps`, `coordinates`, `structure`, `parameters`, `exclude`, `outputname` and one of the following three: `temperature`, `velocities`, `binvelocities`.

NAMD Configuration File for decalanin

```
numsteps 10000                          protocol params

coordinates alanin.pdb                   initial config
temperature 300K
seed 12345

outputname ./alanin                      output params
binaryoutput no

timestep 1.0                         integrator params

structure alanin.psf                 force field params
parameters alanin.params
exclude scaled1-4
1-4scaling 1.0
switching on
switchdist 8.0
cutoff 12.0
pairlistdist 13.5
margin 0.0
stepspercycle 20
```

Gromacs

- GROMACS (Groningen Machine for Chemical Simulations) is a versatile package which performs molecular dynamics, i.e. simulates the Newtonian equations of motion for systems with hundreds to millions of particles.
- It is primarily designed for biochemical molecules like proteins and lipids that have a lot of complicated bonded interactions, but since GROMACS is extremely fast at calculating the nonbonded interactions (that usually dominate simulations) many groups are also using it for research on non-biological systems, e.g. polymers.
- GROMACS calling structure is quite complicated, and it is not recommended that users proceed without at least becoming familiar with the program.
- You will need a protein structure file (pdb file) and a file containing energy minimization data.

Files at /home/apacheco/CompChem/CMD/GROMACS

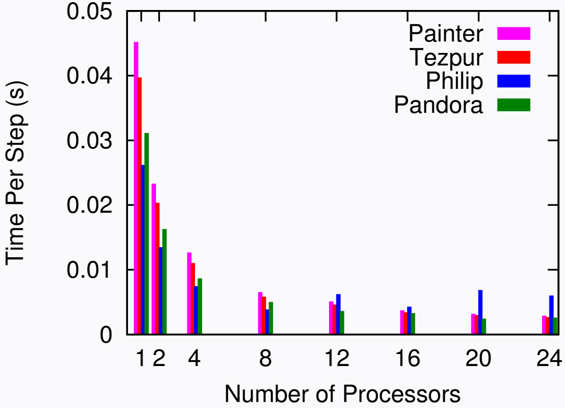
1. Generate topology file using pdb2gmx
```
pdb2gmx_d -f cpeptide.pdb -o cpeptide.gro -p cpeptide.top
```
2. Define the water box size
```
editconf_d -f cpeptide.gro -o cpeptide.gro -d 0.5
```
3. Solvate the peptide
```
genbox_d -cp cpeptide.gro -cs -o cpeptide_b4em.gro -p cpeptide.top
```
4. Energy Minimization, step 1: preprocess the input files.
```
grompp_d -f em -c cpeptide_b4em -p cpeptide -o cpeptide_em
```
5. Energy minimization, step 2
```
mdrun_d -s cpeptide_em -o cpeptide_em -c cpeptide_b4pr -v
```
6. Position Restrained MD, step 1: generate the binary topology file
```
grompp_d -f pr -c cpeptide_b4pr -r cpeptide_b4pr -p cpeptide -o cpeptide_pr
```
7. Position Restrained MD, step 2: run the simulation
```
mdrun_d -s cpeptide_pr -o cpeptide_pr -c cpeptide_b4md -v
```
8. MD Simulation, step 1: generate the binary topolgy file
```
grompp_d -f md -c cpeptide_b4md -p cpeptide -o cpeptide_md
```
9. MD simulation, step 2: run the simulation
```
mdrun_d -s cpeptide_md -o cpeptide_md -c cpeptide_after_md -v
```

Outline

1. Introduction
2. Fundamentals of Molecular Dynamics
3. *Ab Initio* Molecular Dynamics Theory
4. Computational Chemistry Programs
    - Gaussian
    - GAMESS
    - CPMD
    - Amber
    - LAMMPS
    - NAMD
    - Gromacs
5. **Benchmarks**
6. Molecular Dynamics on GPU

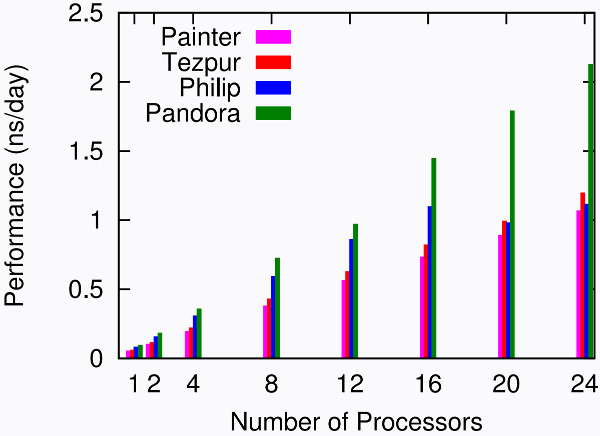
Benchmarks

- LAMMPS on LONI & LSU HPC machines
  - Atomic fluid with Lennard-Jones Potential
  - 32,000 atoms for 100,000 steps
  - force cutoff = 2.5σ, neighbor skin = 0.3σ, neighbor/atom = 55
  - NVE time integration

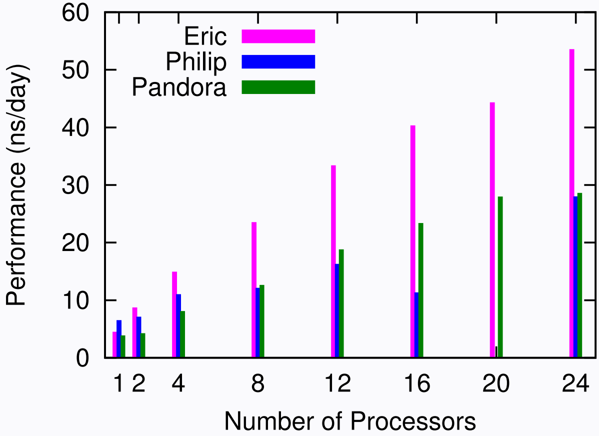


Benchmarks

- NAMD on LONI & LSU HPC machines
  - Apoa1 Benchmark
  - 92,224 atoms, 12Åcutoff + PME every 4 steps, periodic



- Gromacs on LONI & LSU HPC machines
  - 159 residue protein dihydrofolate reductase (dhfr) with implicit solvent
  - 2489 atoms, 1nm cutoff



Outline

1. Introduction
2. Fundamentals of Molecular Dynamics
3. *Ab Initio* Molecular Dynamics Theory
4. Computational Chemistry Programs
   - Gaussian
   - GAMESS
   - CPMD
   - Amber
   - LAMMPS
   - NAMD
   - Gromacs
5. Benchmarks
6. Molecular Dynamics on GPU

Molecular Dynamics on GPU

- Using a graphics processing unit (GPU) for molecular simulations.
- MD programs capable of GPU computing
  1. LAMMPS
  2. NAMD
  3. GROMACS
  4. Amber/PMEMD
- Not all features of the MD programs are capable of GPU computing.

Molecular Dynamics on GPU

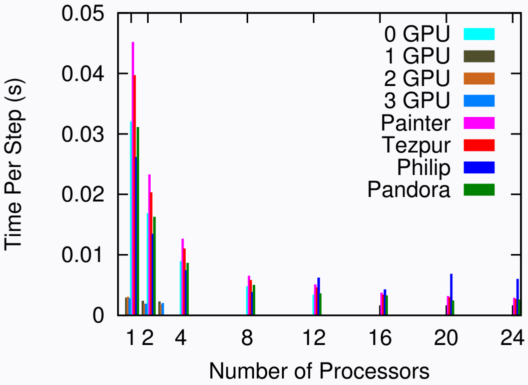
- 2 Philip nodes with 3 GPU’s each.
  - Intel Xeon X5650
    - 2.66GHz Dual Hexa-core with hyperthreading: 12 cores, 24 threads
    - Memory : 48GB
    - L3 Cache: 12MB
  - Tesla M2070
    - 448 CUDA cores (14 Multiprocessor × 32 CUDA Cores/MP)
    - GPU Clock Speed: 1.15GHz
    - Total Memory: 5.25GB
- Compilers: CUDA 4.0, PGI Accelerator
- Infiniband Interconnect between the two nodes.
- For optimum scaling, ratio for CPU core:GPU should be 1.
- LAMMPS and NAMD can bind GPU to multiple cores but performance degrades.

LAMMPS on GPU

- Need to use `newton off`
- `package gpu` should be used in order to initialize and configure the GPUs for use

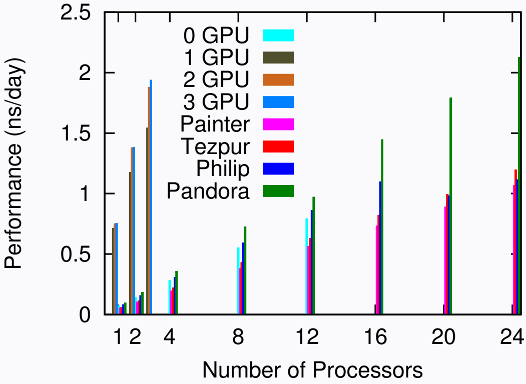
  `package gpu mode first last split`

  `mode` can be force or force/neigh
- currently limited to a few `pair_styles` and PPPM



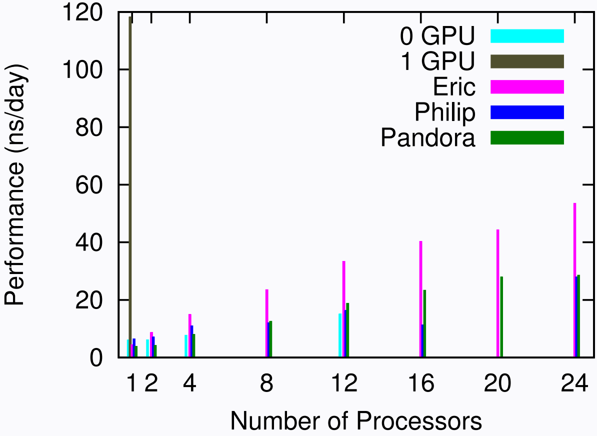
NAMD on GPU

- command line options to add
  `+idlepoll +devices 0,1`
  If `+devices` options is not added then all available GPU's will be used
  Each CPU will be assigned a GPU cyclically
  i.e. `mpirun -np 4 namd2 +idlepoll +devices 0,1` will bind CPU 0 and 2 to GPU device 0 and CPU 1 to GPU device 1



Gromacs on GPU

- Use mdrun compiled with cuda: `mdrun-gpu`
- `mdrun-gpu` can only run on 1 GPU, multiple GPUs functionality not available yet
- Usage: same as serial or parallel mdrun



Related HPC Tutorials

HPC Training in Fall

- Introduction to Electronic Structure Calculation in Quantum Chemistry using Gaussian, GAMESS and NWCHEM.

HPC Training in Spring

- Molecular Dynamics.
- Introduction to Autodock and Autodock Tools. March 28th

LONI HPC Moodle

1. HPC 108: How to use CPMD
2. HPC 109: Intro to Gaussian

Useful Links

- Amber:`http://ambermd.org`
- Desmond:`http://www.deshawresearch.com/resources_desmond.html`
- DL_POLY:`http://www.cse.scitech.ac.uk/ccg/software/DL_POLY`
- Gromacs:`http://www.gromacs.org`
- LAMMPS:`http://lammps.sandia.gov`
- NAMD:`http://www.ks.uiuc.edu/Research/namd`
- CPMD: `http://www.cpmd.org`
- GAMESS: `http://www.msg.chem.iastate.edu/gamess`
- Gaussian: `http://www.gaussian.com`
- NWCHEM: `http://www.nwchem-sw.org`
- PINY_MD:`http://homepages.nyu.edu/~mt33/PINY_MD/PINY.html`
- Basis Set: `https://bse.pnl.gov/bse/portal`

Further Reading

- A Molecular Dynamics Primer by Furio Ercolessi `http://www.fisica.uniud.it/~ercolessi/md`
- Molecular Modeling - Principles and Applications, A. R. Leach.
- Computer Simulation of Liquids, M. P. Allen and D. J. Tildesley.
- Mark Tuckerman's Notes at NYU: `http://www.nyu.edu/classes/tuckerman/quant.mech/index.html`
- Ab Initio Molecular Dynamics: Theory and Implementation, D. Marx and J. Hutter`http://www.theochem.ruhr-uni-bochum.de/research/marx/marx.pdf`
- Quantum Dynamics with Trajectories: Introduction to Quantum Hydrodynamics, R. E. Wyatt.
- Quantum Dynamics of Complex Molecular Systems, Editors: D. A. Micha and I. Burghardt
- Energy Transfer Dynamics in Biomaterial Systems. Editors: I. Burghardt, V. May, D. A. Micha and E. R. Bittner .

Next Week

LONI Programming Workshop, Feb 13-16 in 338 Johnston Hall

Next HPC Training

Introduction to Eclipse Parallel Tools Platform, Feb 29 in 307 Frey CSC