Outline

1. Particle Simulations
   - N-Body Problems
   - Symplectic Integrators
   - Potentials

2. All-Pair Interactions
   - Particle Decomposition
   - Force Decomposition

3. Distance-Limited Interactions
   - Spatial Decomposition
   - Neutral Territory Methods
   - Smooth Particle Mesh Ewald Method
   - Hierarchical Methods
N-Body Problems

Many physical systems can be modeled as collections of interacting particles.

“Particles” vary from atoms in a molecule to planets in a solar system or stars in a galaxy.

Particles exert mutual forces on each other, such as gravitational or electrostatic forces.
N-Body Model

- Newton’s Second Law
  \[ F = m \alpha \]
- Force between particles at positions \( x_i \) and \( x_j \)
  \[ f(x_i, x_j) \]
- Overall force on \( i \)th particle
  \[ F(x_i) = \sum_{j=1}^{n} f(x_i, x_j) \]
N-Body Simulation

- **System of ODEs**

  \[ F(x_i) = m_i \frac{d^2 x_i}{dt^2} \]

- **Verlet** time-stepping scheme

  \[ x_{i}^{k+1} = 2x_{i}^{k} - x_{i}^{k-1} + (\Delta t)^2 F(x_{i}^{k})/m_i \]

- For long time integration, *symplectic* integrators are appropriate (preserve geometric properties, such as orbits)

- Velocity Verlet scheme used in molecular dynamics to preserve energy

- \( O(n^2) \) cost of evaluating force at each time step dominates overall computational cost
Molecular Dynamics

A molecular dynamics simulation performs the following calculations at every *timestep*:

1. Calculate non-bonded forces \( F_{ij} \) for each pair \( (i, j) \) of particles (atoms).
2. Integrate non-bonded forces \( f_i = \sum_j F_{ij} \).
3. Consider local bonded many-particle interactions and update \( f_i \).
4. Update acceleration \( a_i = f_i / m_i \) and velocity \( v_i \) using \( a_i \).
5. Compute new particle position \( x_i \) using \( v_i \) and \( a_i \).
Van der Waals Forces

Short-range atomic interactions governed by electronic coupling (Pauli exclusion principle)

- Molecular bonds typically treated specially
- Short-range ‘non-bonded’ forces modelled by Van der Waals (dipole) potential
- These are based on approximations to the electronic wavefunction
- A popular simple formulation is the Lennart-Jones potential

$$F_{LJ}(x_i, x_j) = \frac{1}{x_i - x_j} \left( \frac{\sigma_{ij}^{(A)}}{|x_i - x_j|^{12}} - \frac{\sigma_{ij}^{(B)}}{|x_i - x_j|^{6}} \right)$$

where $\sigma_{ij}^{(A)}$ and $\sigma_{ij}^{(B)}$ depend on the types of atoms particles $i$ and $j$ are
Electrostatic potentials describe Coulomb’s law for electric fields due to charge

- They decay slowly relative to Van Der Waals interactions

\[
F_{EC}(x_i, x_j) = (x_i - x_j) \frac{q_i q_j}{|x_i - x_j|^3}
\]

where \(q_i\) and \(q_j\) are the charges of particles at \(x_i\) and \(x_j\)

- Coulomb potential interactions are well-approximated using fast solvers
Particle Decomposition

The simplest way to parallelize MD is by *particle decomposition*

- Fine-grained tasks are particles, each processor is assigned $n/p$ of them
- Processors exchange particles in a ring, computing forces from received particles to original $n/p$
- Parallel execution time is
  \[ T_p(n) = O(p\alpha + n\beta + (n^2/p)\gamma) \]
- Memory footprint is minimal $M_p = \Theta(n)$
- Can reduce latency cost by working with larger subsets of particles
Force Decomposition

*Force decomposition* achieves lower communication volume

- Fine-grained tasks are forces, coarse-grained (aggregated) tasks are square blocks of forces
- Assignment/scheduling of aggregated tasks on processors must control for memory usage
- Each processor gets $s \times t$ block $(st = n^2/p)$, accumulates forces for $\min(s, t)$ particles, by streaming in $\max(s, t)$ other particle data
- Memory footprint per processor is $M_p = p \min(s, t)$, time is

$$T_p(s, t) = O \left( \frac{\max(s, t)}{\min(s, t)} \alpha + \max(s, t) \beta + st \gamma \right)$$

Michael T. Heath and Edgar Solomonik
Algorithms for All-pairs Force Calculation

- 1D – particle decomposition \((c = 1, s = n/p, t = n)\)
- 2D – force decomposition \((c = \sqrt{p}, s = n/\sqrt{p}, t = n/\sqrt{p})\)
- 1.5D – memory-constrained force decomposition
  \((M_p = cn^2, s = cn/p, t = n/c)\)
Molecular dynamics is typically done without explicitly computing all particle interactions

- Van der Waals interactions decay very rapidly and can be ignored for far-away particles

- Electrostatic forces can be computed by fast solvers
  - Electrostatic potential obeys the Poisson equation
  - The gravitational potential (used for cosmological simulation) is also Poisson
  - While pairwise interactions decay slowly, the aggregate potential due to long-range forces will be a smooth function
For molecular dynamics, interactions decoupled as follows

- Compute Van der Waals interactions of all particle pairs \((i, j)\) within distance \(|x_i - x_j| \leq r_c\)

- Fit a 3D charge density grid to the particle charges

- Solve the 3D Poisson equation on the grid via 3D FFT or Multigrid to obtain potential at grid-points

- Extrapolate potential from grid to compute electrostatic forces on particles
  - Force is given by the spatial gradient of potential
  - \textit{B-splines} provide a basis with compact spatial support and easy computation of derivatives
Spatial Decomposition

Domain is $n^{1/3} \times n^{1/3} \times n^{1/3}$ box with uniform density

- MD simulations are typically done inside ‘solute’ (water), and have uniform density
- Uniform density does not necessarily hold in other domains, e.g. cosmological simulations
- Fine-grained tasks are unit-volume boxes
- Aggregated-tasks (boxes) are mapped to processors
- Each processor can have subdomain of dimensions $(n/p)^{1/3} \times (n/p)^{1/3} \times (n/p)^{1/3}$
- To compute forces onto all these particles, need all particles within $r_c$ away from subdomain
  $$W_p(n, r_c) = O((r_c + (n/p)^{1/3})^3 - n/p) = O(r_c^3 + r_c(n/p)^{2/3})$$
Neutral Territory Methods

Spatial decomposition leverage locality of particles, *neutral territory methods* directly exploit locality of forces

- Allow interactions between particles owned by two different processors to be computed on a third, in *neutral territory*
3D Neutral Territory Methods

Minimal Import Regions

Assign each processor $k$ is assigned a unique subvolume $X_k \times Y_k \times Z_k$ of dimensions $b_{xy} \times b_{xy} \times b_z$ such that $b_{xy}^2 b_z = n/p$

- Processor $k$ computes interactions of particle pair $(i, j)$ if
  - $i$ and $j$ have a $z$-coordinate in $Z_k$ and $x, y$-coordinates within $r_c$ of some element in $X_k, Y_k$, respectively
  - $i$ and $j$ have $x, y$-coordinates in $X_k, Y_k$ and a $z$-coordinate within $r_c$ of some element in $Z_k$

- The volume of the region (amount of communication) is
  \[
  W_p(n, r_c, b_{xy}, b_z) = O(r_c b_{xy}^2 + r_c b_z b_{xy} + r_c^2 b_z)
  \]

- Minimizing the import region with respect to $b_{xy}$ and $b_z$
  \[
  W_p(r_c) = O(r_c(n/p)^{2/3} + \sqrt{r_c^3 n/p})
  \]
Smooth Particle Mesh Ewald (SPME)

Solve for long range interactions on a $m \times m \times m$ charge grid

- System assumed periodic, which is often valid in MD
- *Ewald summation* is used to split the total potential energy

\[
E = \frac{1}{2} \sum_{c \in \mathbb{Z}^3} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{q_i q_j}{|x_i - x_j + cn^{1/3}|}
\]

into two parts (the form here is slightly simplified)
Ewald Summation

- The first part is a dampened direct summation

\[
E_{\text{dir}} = \frac{1}{2} \sum_{c \in \mathbb{Z}^3} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{q_i q_j \text{erfc}(\beta |x_i - x_j + cn^{1/3}|)}{|x_i - x_j + cn^{1/3}|}
\]

the function \(\text{erfc}(y)\) is the probability a uniform random variable with mean 0 and variance 1/2 falls outside of the range \([-y, y]\), so pairs with sufficiently large \(x_i - x_j\) or in distant cells can be ignored

- The reciprocal (second part) is a convolution over all charge grid cells, except \(c = (0, 0, 0)\) is contracted based on \(\beta\)
The forces on particles in SPME are obtained by equations that are derivatives of the energy with respect to position

- **SPME with** \( m \times m \times m \) **grid calculates the reciprocal portion as follows**
  - B-splines interpolate charge from nearby region of particles
    \[
    T_p(n, m) = O(\alpha + (n/p)^{2/3} \beta + (m^3/p)\gamma)
    \]
  - The grid convolution by 3D FFT for \( p \leq m^{5/2} \) takes time
    \[
    T_p(m) = O(\log p \alpha + (m^3/p) \beta + (m^3 \log(m)/p)\gamma)
    \]
  - Extrapolating potential from grid to particles
    \[
    T_p(m) = O(\log p \alpha + (m^2/p^{2/3}) \beta + (m^3/p)\gamma)
    \]
Alternative Methods

- Poisson equation on grid can theoretically be solved fastest by multigrid
- SPME can outperform multigrid in practice, achieving high accuracy with a small grid
- Advantage in part due to sensibility of periodicity condition
- Particle simulations with unbalanced particle distributions require different methods
- The *Barnes-Hut* method and the *Fast Multipole Method* (*FMM*) leverage hierarchical domain partitioning
Tree Partitioning for N-Body Problems

Tree-based methods such as Barnes-Hut and FMM replace a set of forces from far-away particles with a single aggregate approximate force.
Barnes-Hut simulations provide a hierarchical spatial decomposition suitable for unbalanced distributions.

Subdivide space recursively until cells contain $O(k)$ particles:
- In 1D, obtain a binary tree.
- In 2D, obtain a quad tree.
- In 3D, obtain an oct tree.

Compute a centered mass/charge for each tree node or in terms of a Taylor series for higher accuracy.

Calculate forces between far-away particles in far-away cells, based on interaction with particle and a mass/charge at a higher-level tree node.
Barnes-Hut

Diagram taken from course webpage of Mowry and Railing (CMU)
Fast Multipole Method (FMM)

FMM obtains linear complexity for integral equations

- Derivations specific to equations, Greengard and Rokhlin originally focused on 2D electrostatics
- In Barnes-Hut leaves interact with tree nodes, in FMM, tree nodes interact with $O(1)$ other tree nodes
- Each node has a multipole (inner) and Taylor (outer) expansion consisting of $O(\log(1/\epsilon))$ terms for accuracy $\epsilon$
  - Error is controlled by number of terms in expansion
  - A multipole expansion is a special type of Taylor expansion
- Transformation operators are defined to ‘shift’ multipole and Taylor expansions, and to convert between the two
FMM Algorithm

The computation in FMM proceeds as follows

1. Perform interactions among particles in neighboring blocks
2. Upward pass – generate multipole expansion for every tree node starting from leaves
3. Downward pass – generate local expansion for every tree node starting from root

Structure and execution time model is analogous to HSS matrices, but with some differences

1. All neighboring cells interact directly
2. Amount of work associated with each tree node may vary
References - Particle Simulations

References - Particle Simulations

References - Parallel Particle Simulations

- M. Driscoll et al., A communication-optimal n-body algorithm for direct interactions, IPDPS, Boston, May 2013
