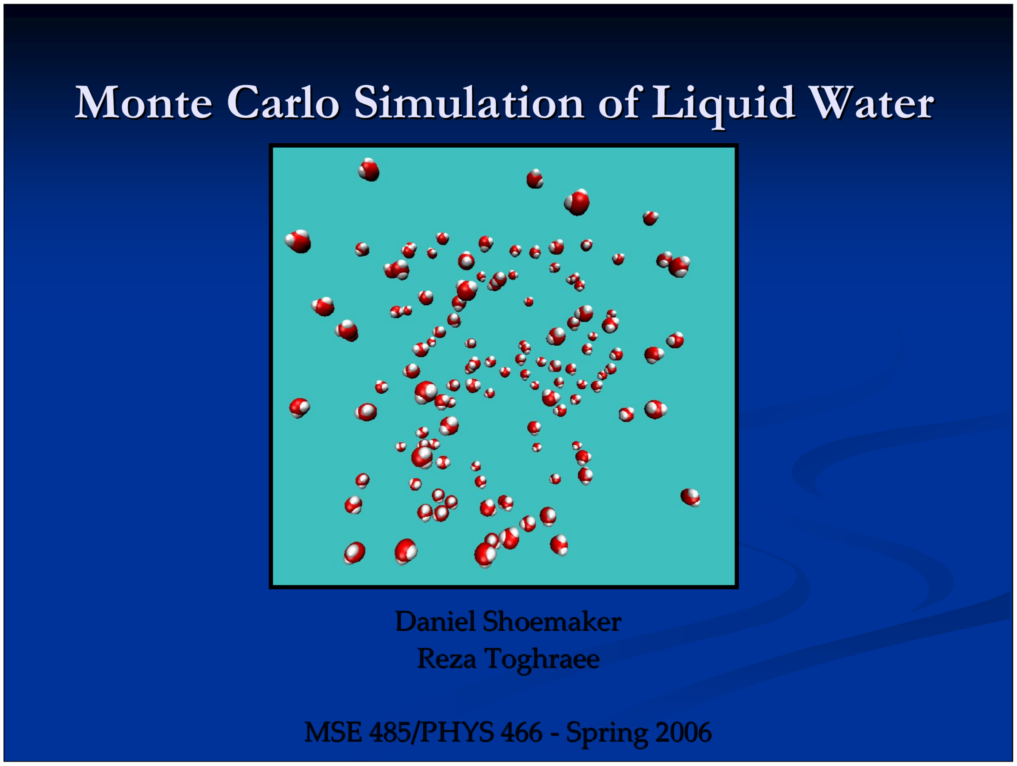
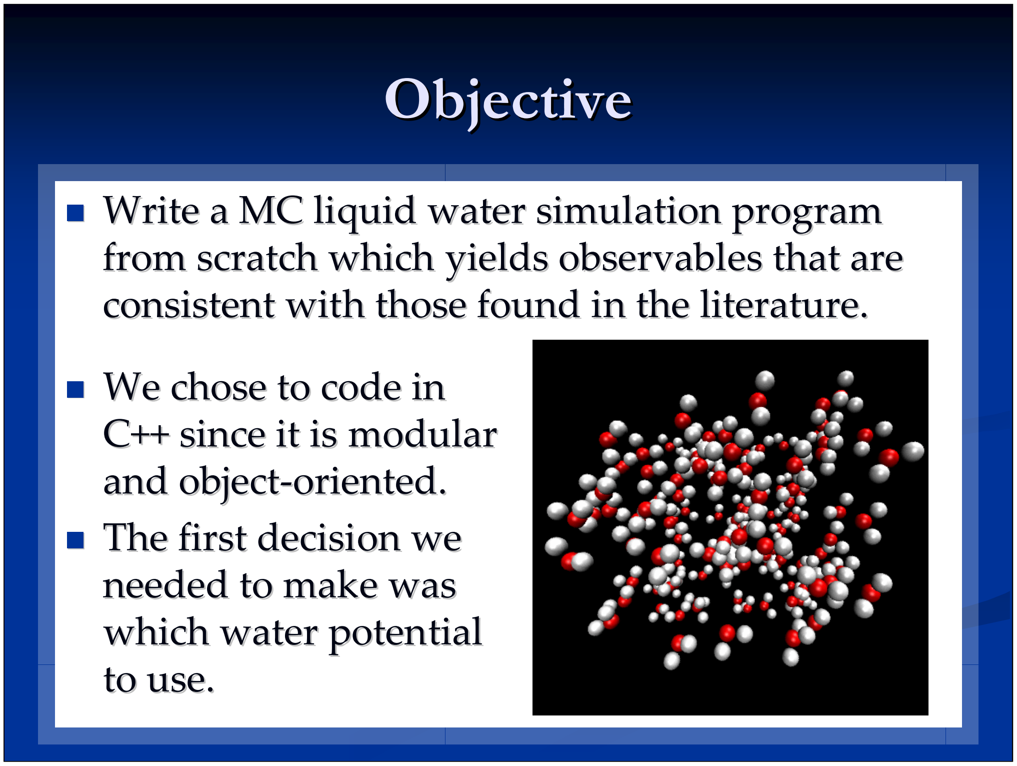


Monte Carlo Simulation of Liquid Water

Daniel Shoemaker
Reza Toghraee

MSE 485/PHYS 466 - Spring 2006

created using
BCL easyPDF
Printer Driver

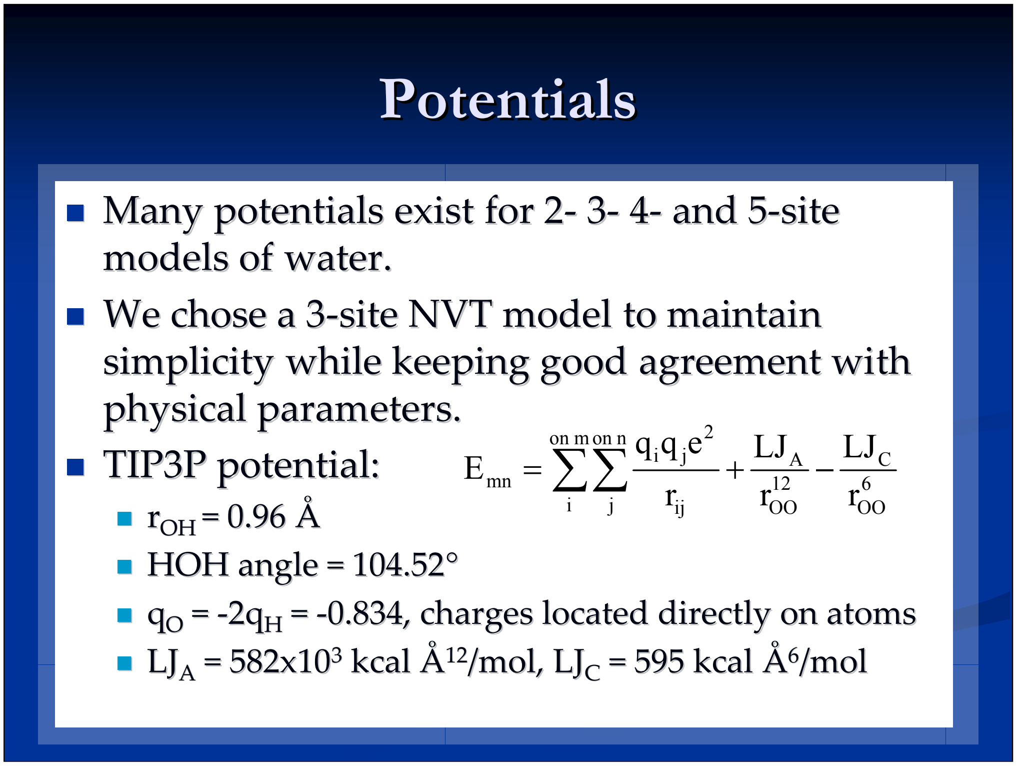


Objective

- Write a MC liquid water simulation program from scratch which yields observables that are consistent with those found in the literature.
- We chose to code in C++ since it is modular and object-oriented.
- The first decision we needed to make was which water potential to use.

The animated gif shows positions of 125 water molecules every 1000 steps in the last 200K iterations of a 5 million iteration process.

created using
BCL easyPDF
Printer Driver



Potentials

- Many potentials exist for 2- 3- 4- and 5-site models of water.
- We chose a 3-site NVT model to maintain simplicity while keeping good agreement with physical parameters.
- TIP3P potential:

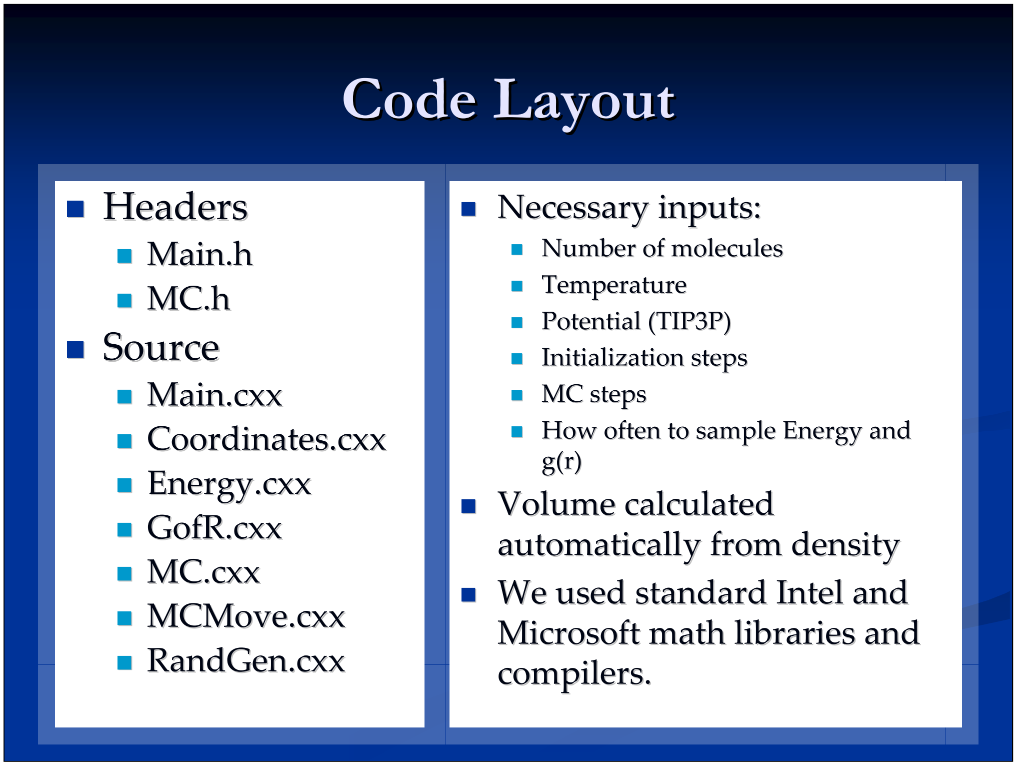
$$E_{mn} = \sum_{i}^{\text{on } m}\sum_{j}^{\text{on } n}\frac{q_i q_j e^2}{r_{ij}} + \frac{LJ_A}{r_{OO}^{12}} - \frac{LJ_C}{r_{OO}^{6}}$$

  - r_{OH} = 0.96 Å
  - HOH angle = 104.52°
  - $q_O = -2q_H = -0.834$, charges located directly on atoms
  - $LJ_A = 582\text{x}10^3$ kcal $Å^{12}$/mol, $LJ_C = 595$ kcal $Å^6$/mol

The TIP3P potential used by Jorgensen et al uses the NPT ensemble since it most closely resembles laboratory conditions. It also provides the opportunity to calculate some statistical averages that cannot be found with NVT. We decided on NVT since it was simpler to code and the differences in this project would not be significant.

The TIP3P potential consists of a Lennard-Jones potential for oxygen-oxygen interactions only, with a Coulombic interaction for all O and H atoms, which results in 1 O-O, 4 O-H, and 8 H-H interactions per pair of molecules.

created using
BCL easyPDF
Printer Driver



Code Layout

- Headers
  - Main.h
  - MC.h
- Source
  - Main.cxx
  - Coordinates.cxx
  - Energy.cxx
  - GofR.cxx
  - MC.cxx
  - MCMove.cxx
  - RandGen.cxx

- Necessary inputs:
  - Number of molecules
  - Temperature
  - Potential (TIP3P)
  - Initialization steps
  - MC steps
  - How often to sample Energy and g(r)
- Volume calculated automatically from density
- We used standard Intel and Microsoft math libraries and compilers.

The volume of our simulation is constant, and is computed automatically by the program based on the number of molecules and the density of water at the simulation temperature.

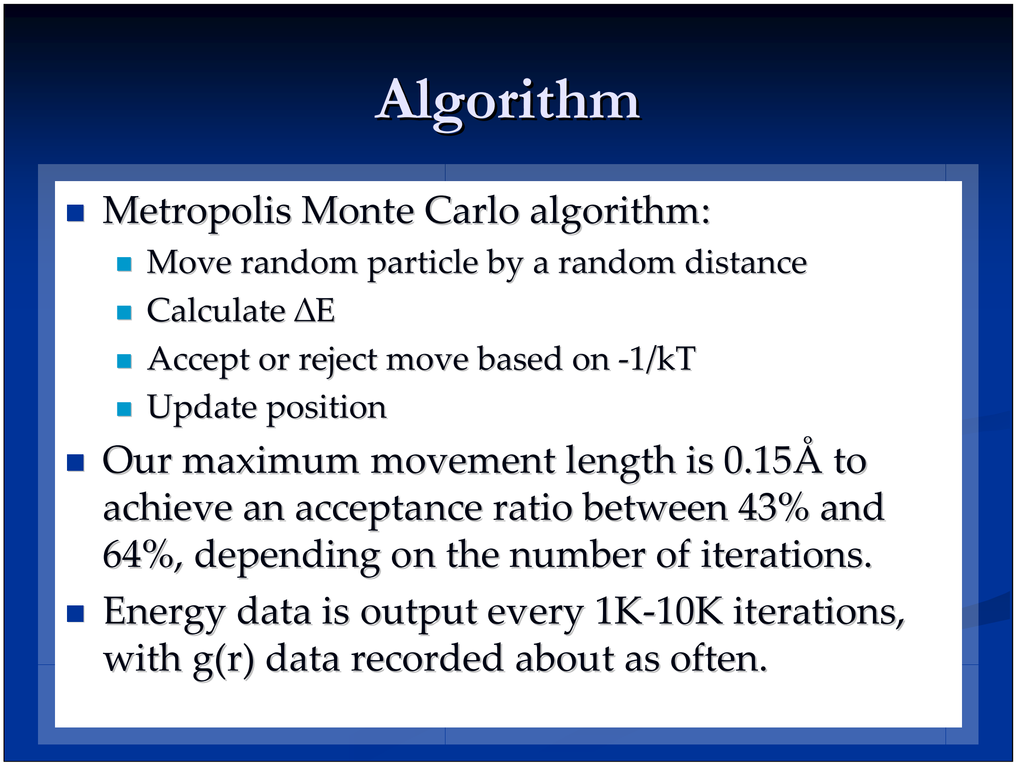
Header files are used to store all of the global variables that define the environment and simulation space as well as the molecule positions and almost any information (cumulative energy, g(r) histogram, etc) that is retained from step to step.

Only local variables that are used within an operation and discarded are defined within the individual source files.

Standard math library random number generators were used, but if more sophisticated methods were desired, the RandGen.cxx could easily be updated with more robust subroutines without any intensive changes to the program as a whole.

Initialization was usually run for ~10K iterations, because the particles are initially distributed randomly in the box without checking for overlaps. Any such anomalies are assumed to be worked out of the system by the favorable energy change associated with breaking them up.

created using
BCL easyPDF
Printer Driver

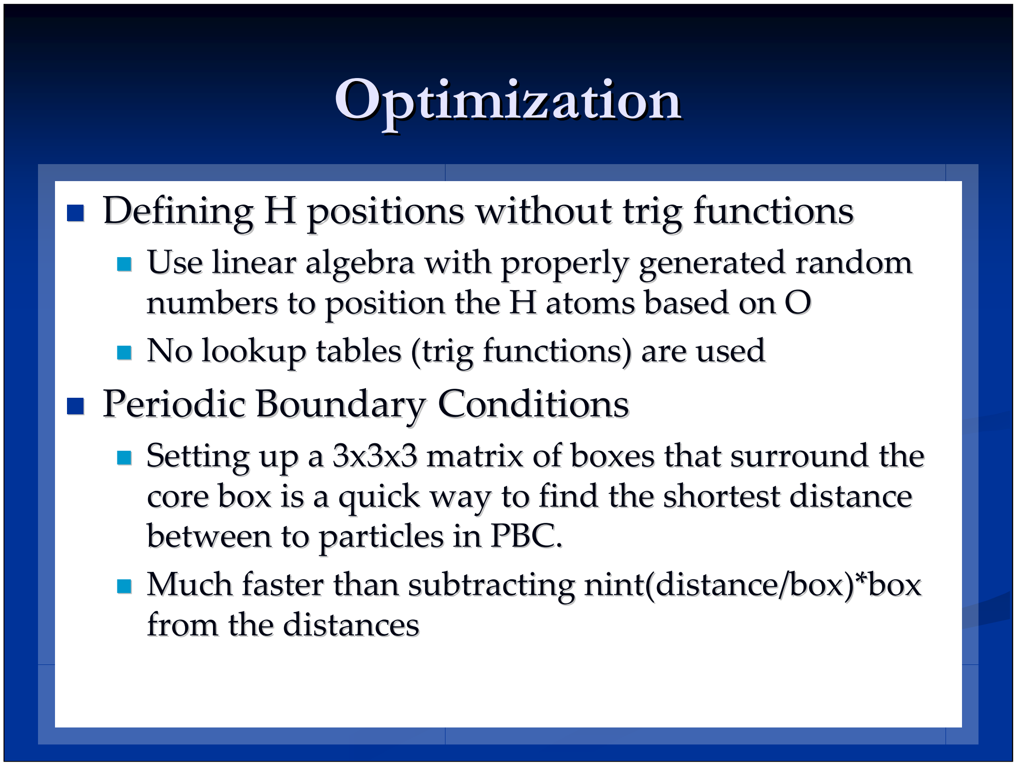


Algorithm

- Metropolis Monte Carlo algorithm:
  - Move random particle by a random distance
  - Calculate ΔE
  - Accept or reject move based on -1/kT
  - Update position
- Our maximum movement length is 0.15Å to achieve an acceptance ratio between 43% and 64%, depending on the number of iterations.
- Energy data is output every 1K-10K iterations, with g(r) data recorded about as often.

The maximum 0.15A jump distance was taken from Jorgensen et al, who achieved ~40% acceptance. For simulations under 100K iterations, our acceptance ratio was ~43%, but the ratio increases to ~64% above 1 million iterations.

created using
BCL easyPDF
Printer Driver



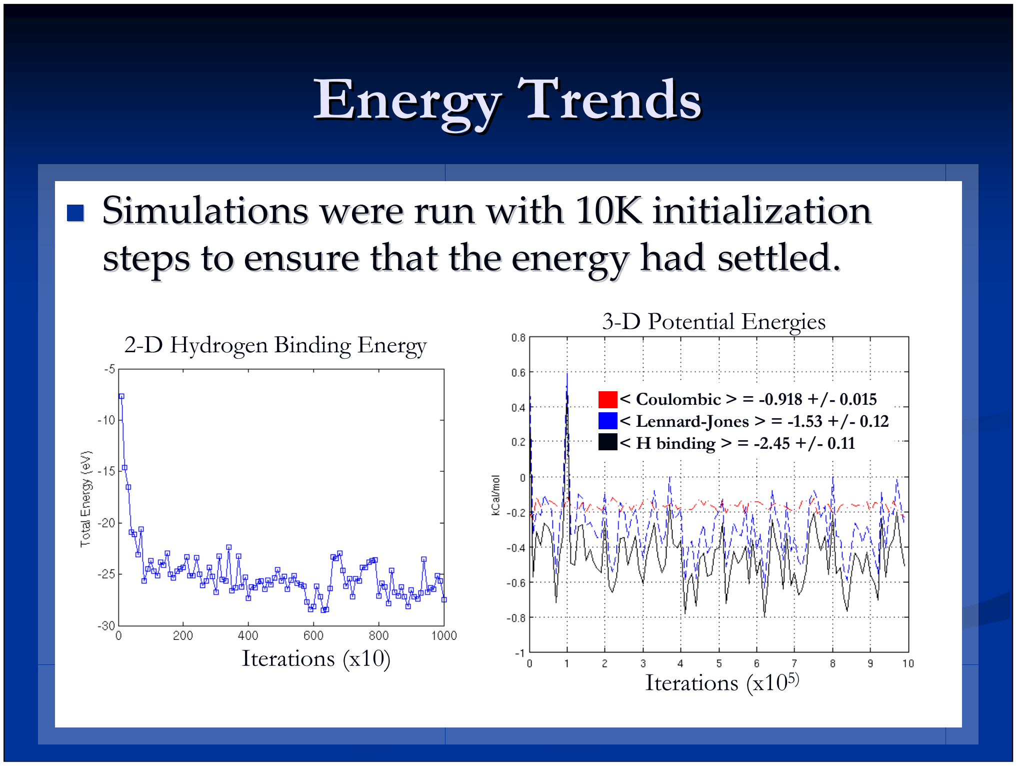
Optimization

- Defining H positions without trig functions
  - Use linear algebra with properly generated random numbers to position the H atoms based on O
  - No lookup tables (trig functions) are used
- Periodic Boundary Conditions
  - Setting up a 3x3x3 matrix of boxes that surround the core box is a quick way to find the shortest distance between to particles in PBC.
  - Much faster than subtracting nint(distance/box)*box from the distances

The H optimization occurs any time a particle is moved, and many times during the initialization of the positions.

The PBC optimization runs any time the distances between particles must be calculated, so it saves a lot of computing time when calculating the energy after every iteration.

created using
BCL easyPDF
Printer Driver



Energy Trends

- Simulations were run with 10K initialization steps to ensure that the energy had settled.

3-D Potential Energy Averages:

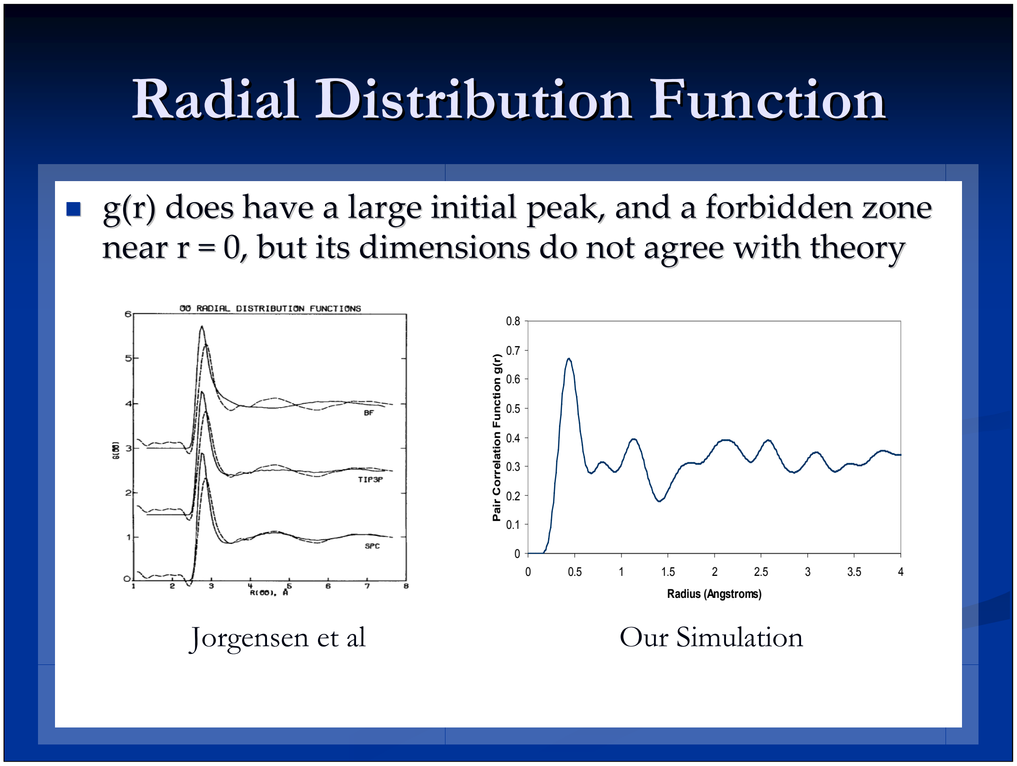
< H binding > = -2.45 +/- 0.11

< Coulombic > = -0.918 +/- 0.015

< Lennard-Jones > = -1.53 +/- 0.12

According to Jorgensen, the LJ energy should be on the order of 1.2 kcal/mol, while the Coulombic energy is about -5.39kcal/mol. That equates to a total H bonding energy of -4.20kcal/mol, or about 3.5 bonds per molecule. Hydrogen bonds are defined by interaction energy of -2.25 kcal/mol or less.

created using
BCL easyPDF
Printer Driver



Radial Distribution Function

- g(r) does have a large initial peak, and a forbidden zone near r = 0, but its dimensions do not agree with theory

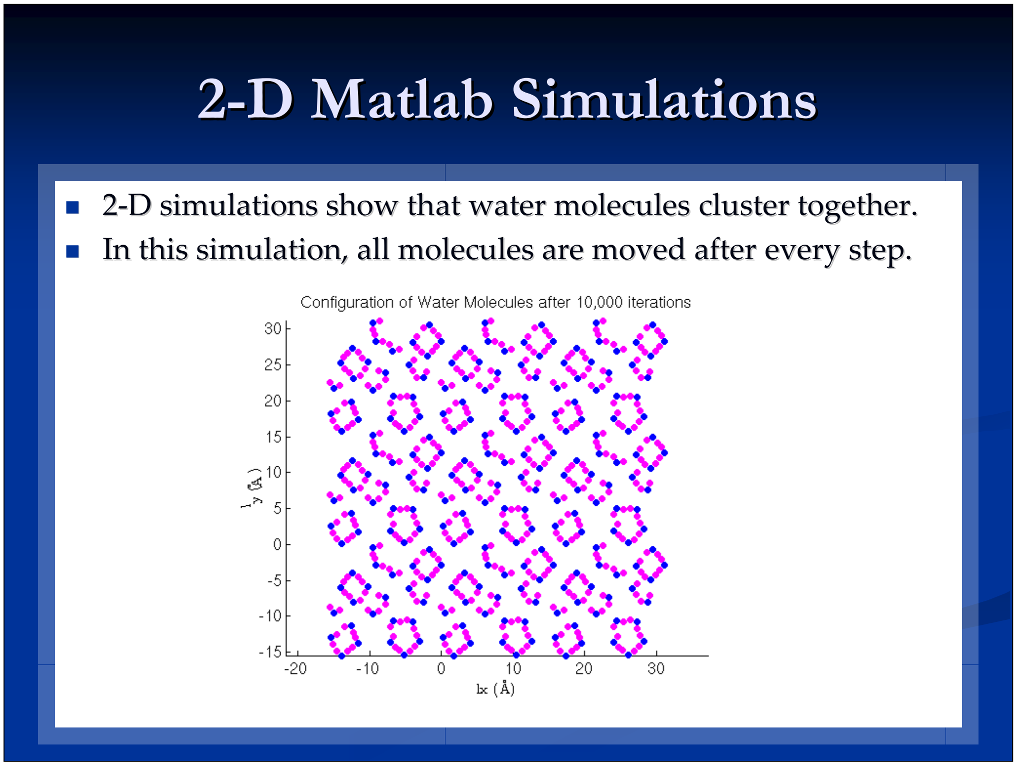
Jorgensen et al

Our Simulation

The scale of our g(r) curve is off, but the general shape is correct. Most likely, there is an incorrect factor in the normalization code that is skewing the size but not distorting the shape. This particular g(r) curve was the result of a 125 particle simulation after about 1000 iterations with binning in the g(r) histogram at 100 step intervals.

Another possible problem is the high LJ potential in our energy calculations, which might have a drastic effect on the close-range interactions of the molecules.

created using
BCL easyPDF
Printer Driver



2-D Matlab Simulations

- 2-D simulations show that water molecules cluster together.
- In this simulation, all molecules are moved after every step.

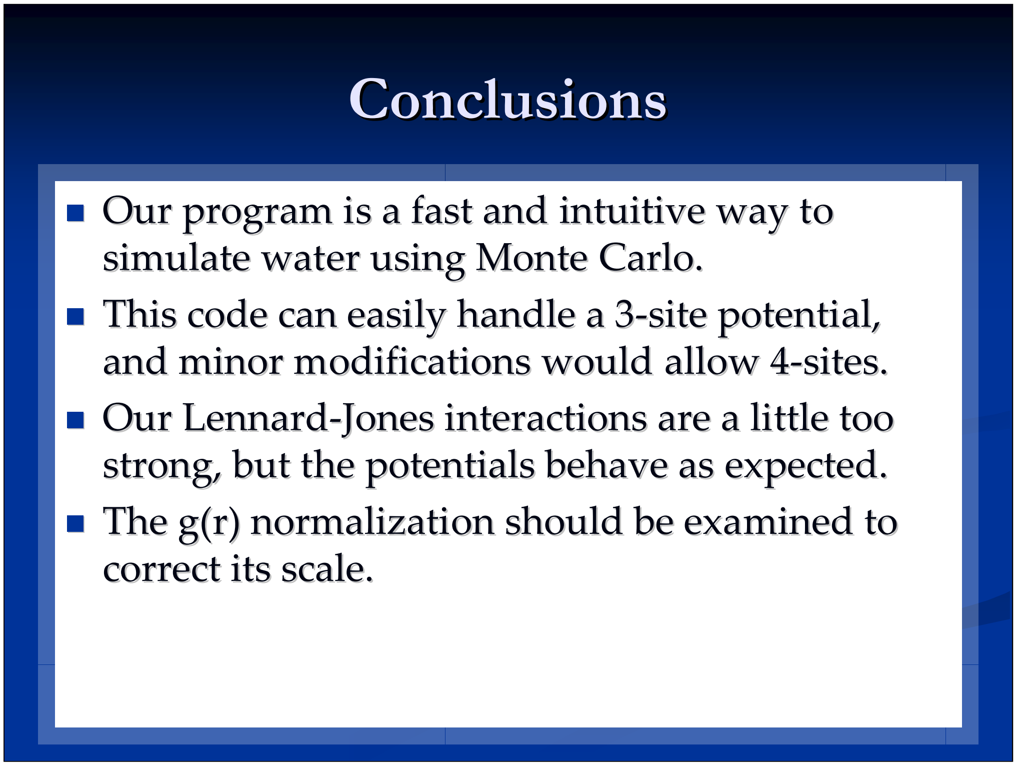
Configuration of Water Molecules after 10,000 iterations

In nature, water molecules cluster together with hydrogen bonding, which is the result of O-H interactions between neighboring molecules.

In this simulation, molecules began in a random distribution with random orientations, and all molecules are moved randomly at each iteration. They curiously bundle together by touching H-H after 10K steps.

This simulation was coded and performed by Jihan Kim.

created using
BCL easyPDF
Printer Driver



Conclusions

- Our program is a fast and intuitive way to simulate water using Monte Carlo.
- This code can easily handle a 3-site potential, and minor modifications would allow 4-sites.
- Our Lennard-Jones interactions are a little too strong, but the potentials behave as expected.
- The g(r) normalization should be examined to correct its scale.

created using
BCL easyPDF
Printer Driver

References

- 2-D Simulations by Jihan Kim

- Berendsen, H. J. C. et al, *Intermolecular Forces*, (D. Reidel Co., Holland 1983), 331.
- Frenkel, D. and B. Smit, *Understanding Molecular Simulation*, (2nd Ed., Academic Press 2002).
- Jorgensen, W. L., *J. Am. Chem. Soc.* **103**, 1981, 335.
- Jorgensen, W. L. et al, *J. Chem. Phys.* **79** (2), 12 July 1983, 926.
- McDonald, I. R., *Mol. Phys.* **23**, 1972, 41.

created using
BCL easyPDF
Printer Driver