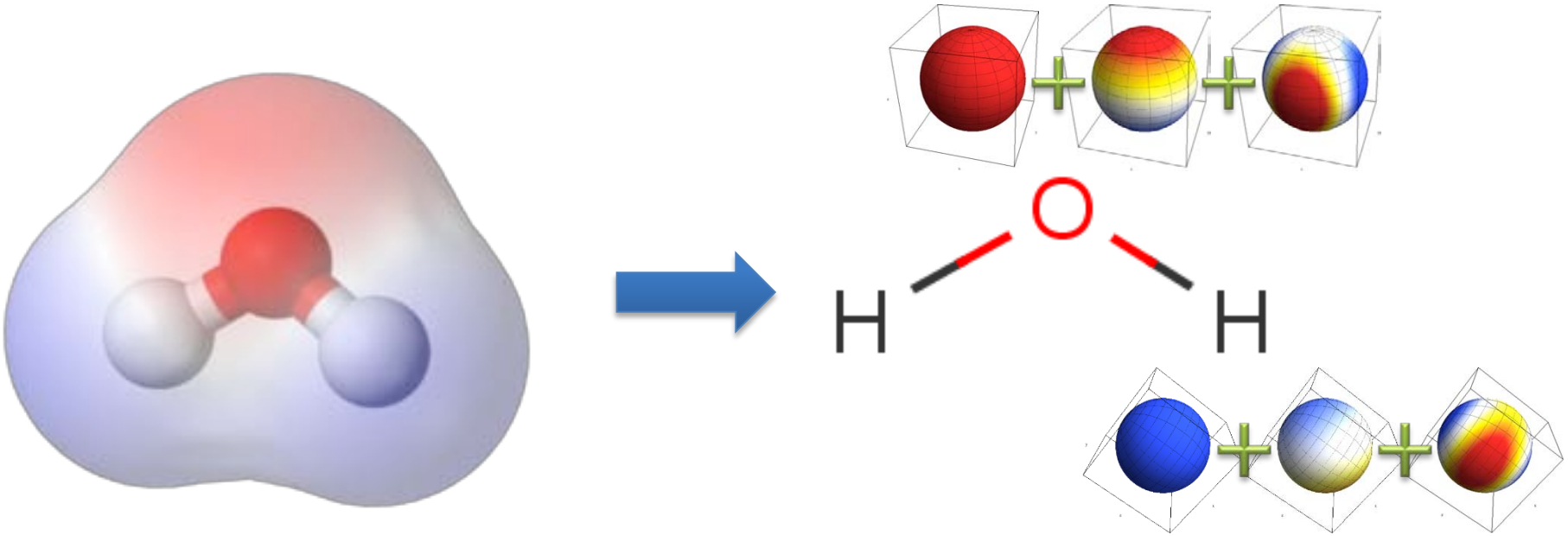
The AMOEBA Polarizable Force Field in OpenMM

Lee-Ping Wang
Stanford Department of Chemistry
OpenMM Workshop, Stanford University
September 6, 2012

Outline

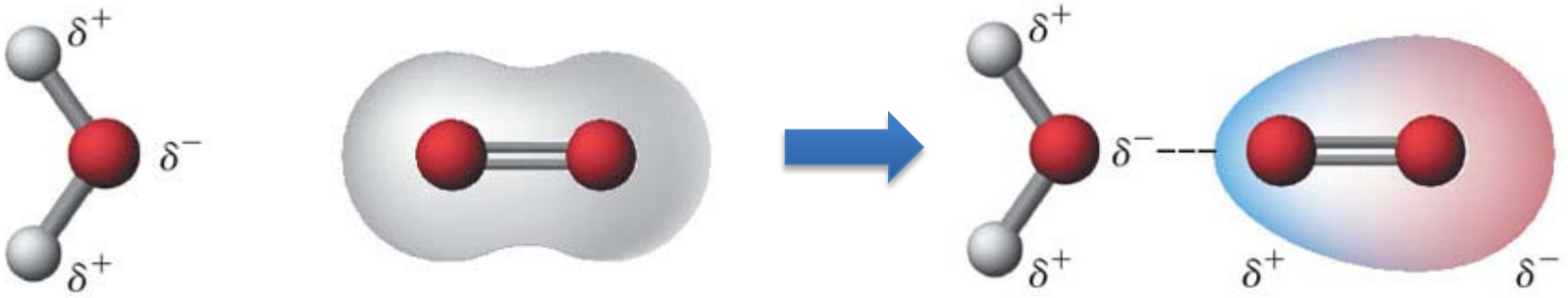
- **Part I: Introduction to the AMOEBA force field**
  - Physical foundations of the AMOEBA force field
  - Validation and comparison to fixed-charge force fields
- Part II: Implementation of AMOEBA in OpenMM
  - Accuracy and performance benchmarks
  - How to set up a simulation



Overview of polarizable force fields

Electronic polarization is a critical part of how a molecule responds to its environment.

Polarization is responsible for *(ion / dipole)-induced dipole* forces



- Electronic polarization happens almost instantly (faster than time step)
- Most force fields use fixed point charges and *cannot describe polarization*
- Point charges are increased from their intrinsic values to describe "average" polarization in the environment
- Thus, point-charge force fields are inherently inaccurate when transferred to different environments! (e.g. water in bulk vs. near a protein)

Introducing the AMOEBA force field

AMOEBA is one of several force fields with the ability to describe electronic polarization.

Some flavors of polarizable force fields:

- Charge-on-spring (a.k.a. Drude) models
- Fluctuating-charge models
- Polarizable point dipoles

AMOEBA belongs to the 3rd category and stands for:

Atomic Multipoles Optimized Energetics for Biomolecular Applications

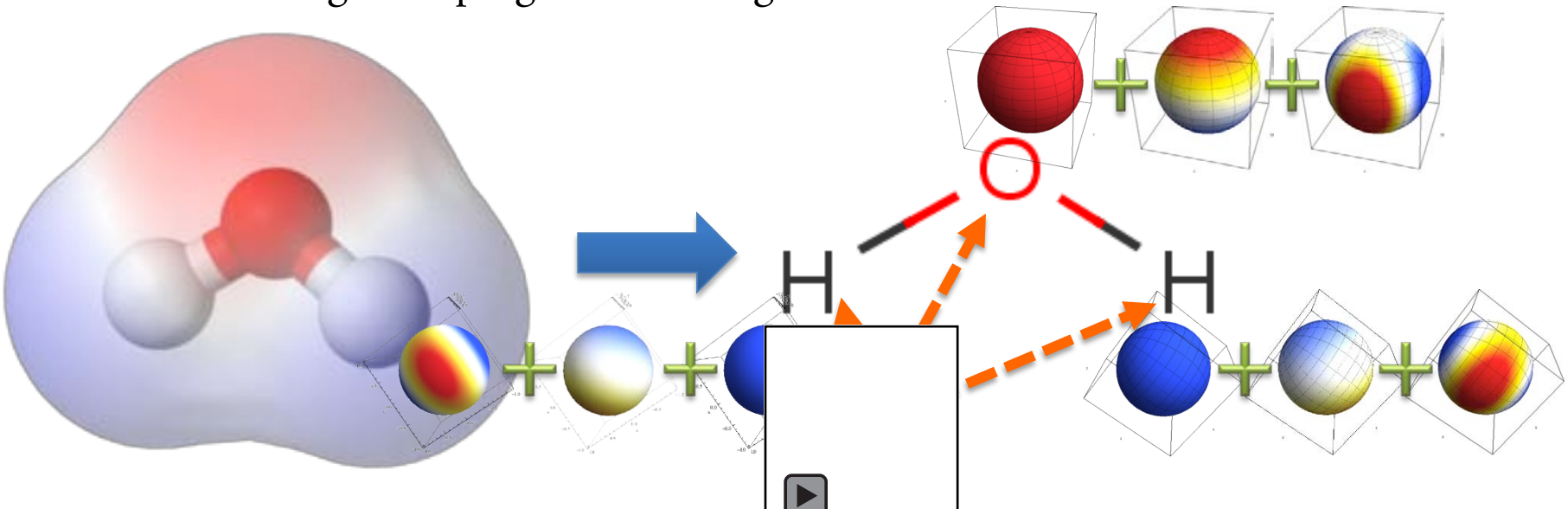
- Developed mainly by Pengyu Ren and Jay Ponder
- Original implementation in TINKER software
- Optimized implementation in OpenMM (topic of this talk)

Polarizable atomic multipoles in AMOEBA

AMOEBA treats electrostatics using a combination of fixed and polarizable multipoles.

Difference between AMOEBA and fixed charge force fields:

- Conventional force fields have point charges on each atom
- AMOEBA puts a point charge, dipole, and quadrupole on each atom
  - Each atom needs to have its own coordinate system
- AMOEBA also adds a polarizable dipole to each atom
  - Short-range damping to avoid singularities



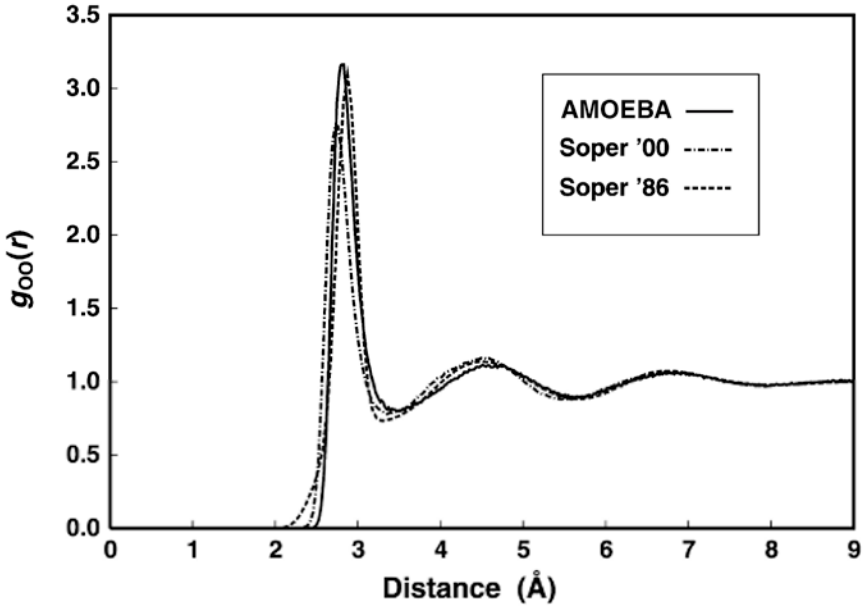
Summary of interactions in AMOEBA

All interaction terms (van der Waals, torsions, etc.) are treated at a higher level of detail.

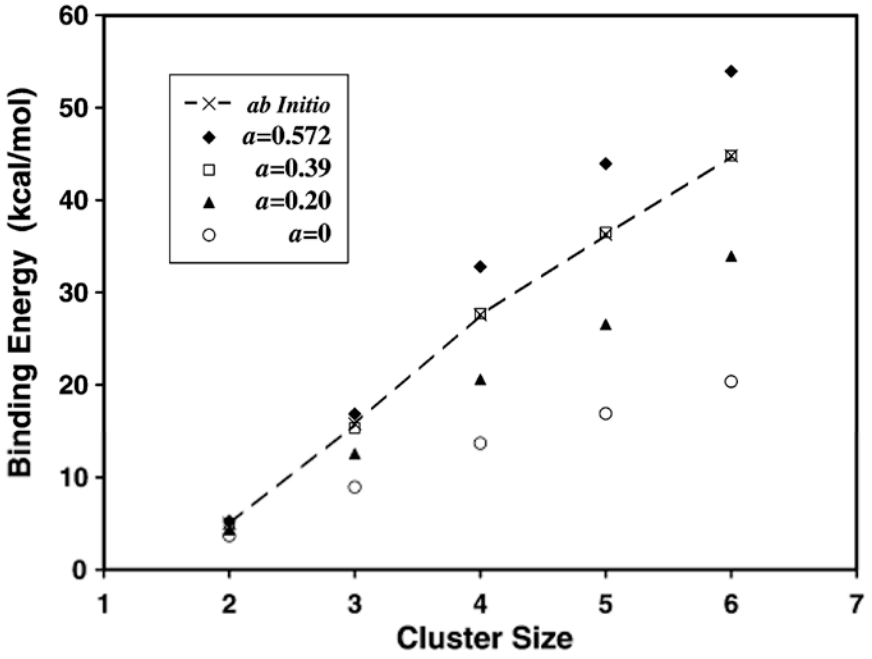
The AMOEBA water model

The AMOEBA water model is capable of describing water in both vapor and liquid phases.

O–O radial distribution function compared to experiment



Binding energy of water clusters (perfect agreement with x's and squares)

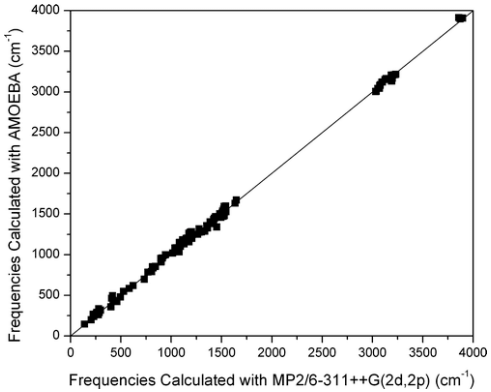


Ren, and Ponder, *J. Phys. Chem. B* 2003, **107**, 5933.

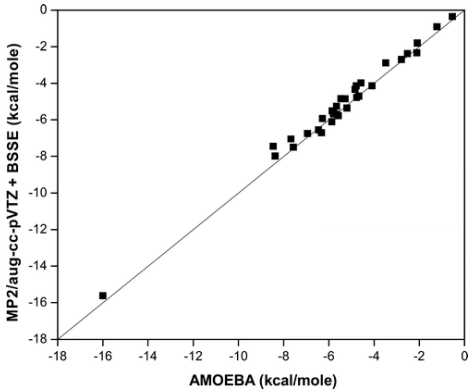
Organic molecules in the AMOEBA force field

A force field for organic molecules has been developed with some validation in the literature.

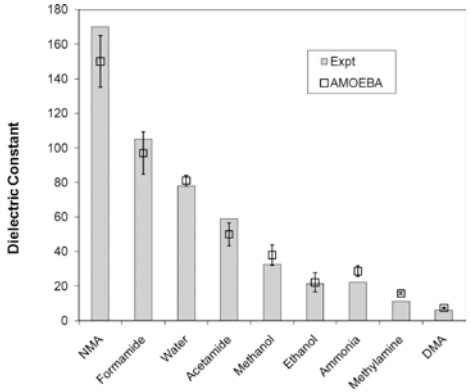
Vibrational frequencies



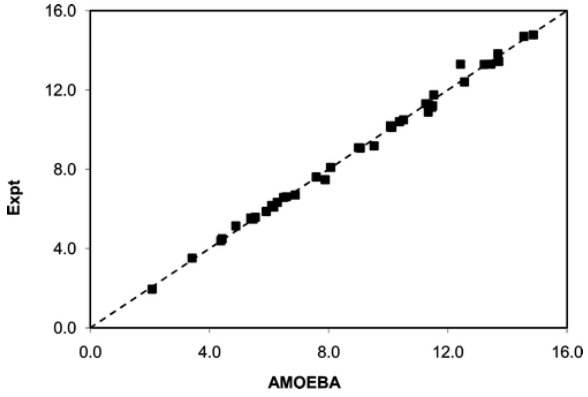
Dimer binding energies



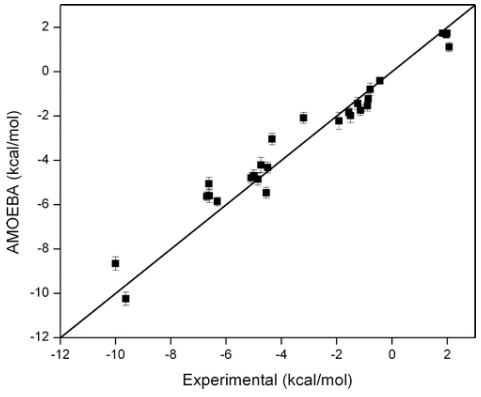
Dielectric constants



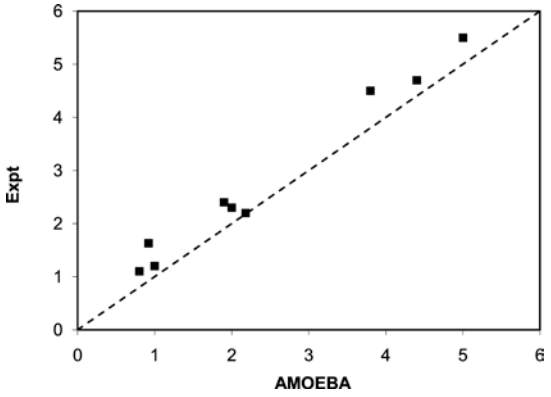
Heats of vaporization



Free energies of solvation



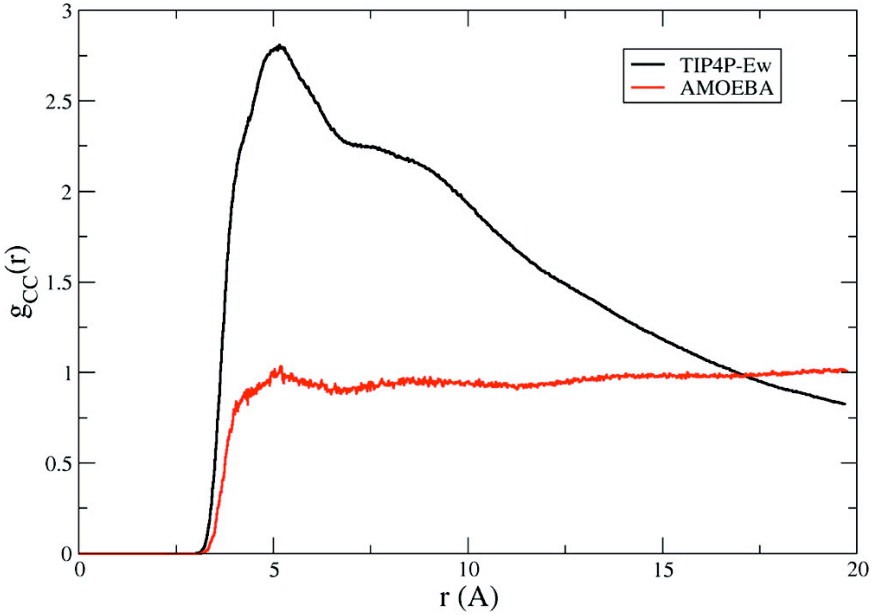
Diffusion constants



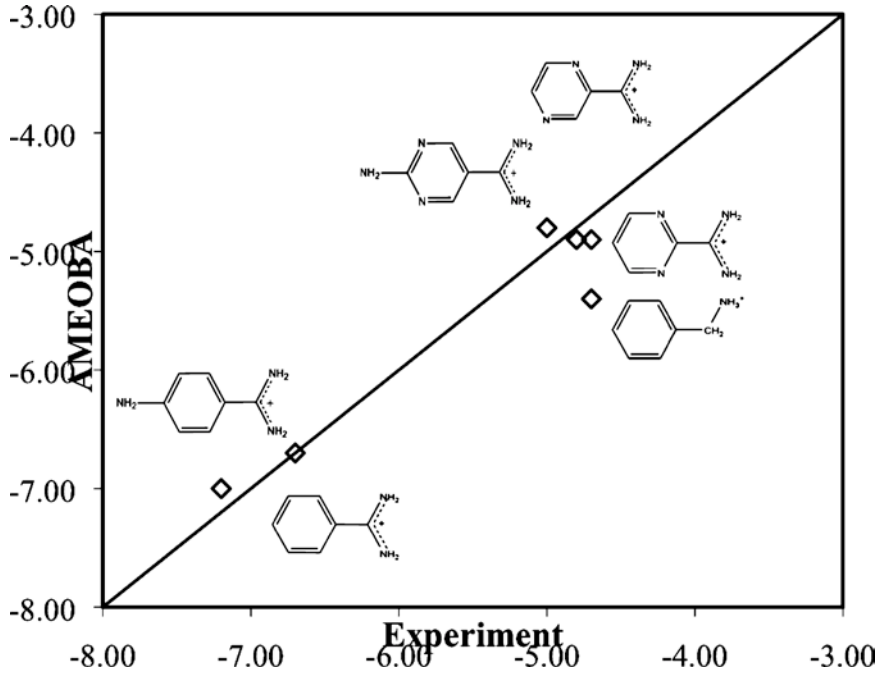
Ren, Wu and Ponder, *J. Chem. Theory Comput.* 2011, **7**, 3143.

Proteins in the AMOEBA force field

A force field for proteins has also been developed with some validation.



Point-charge force field (black) wrongly predicts peptide aggregation, AMOEBA (red) gives correct result

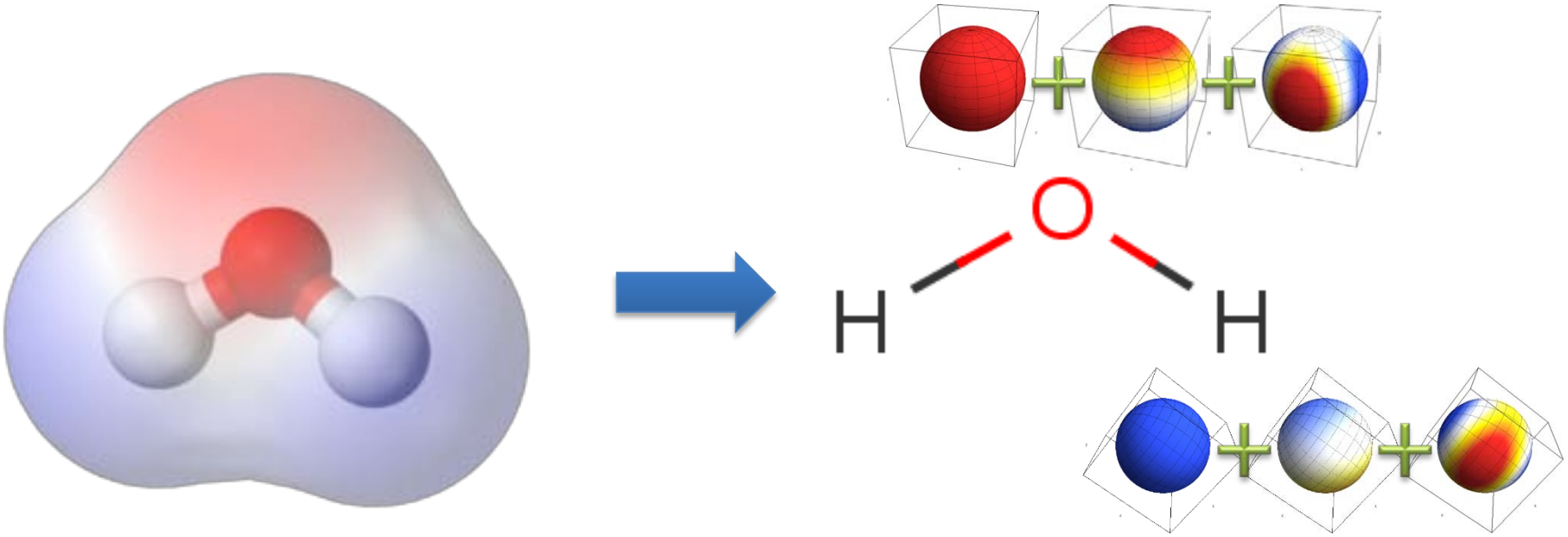


Protein–ligand binding free energies

Ponder, Pande, Chodera et al., *J. Phys. Chem. B* 2010, **114**, 2549.

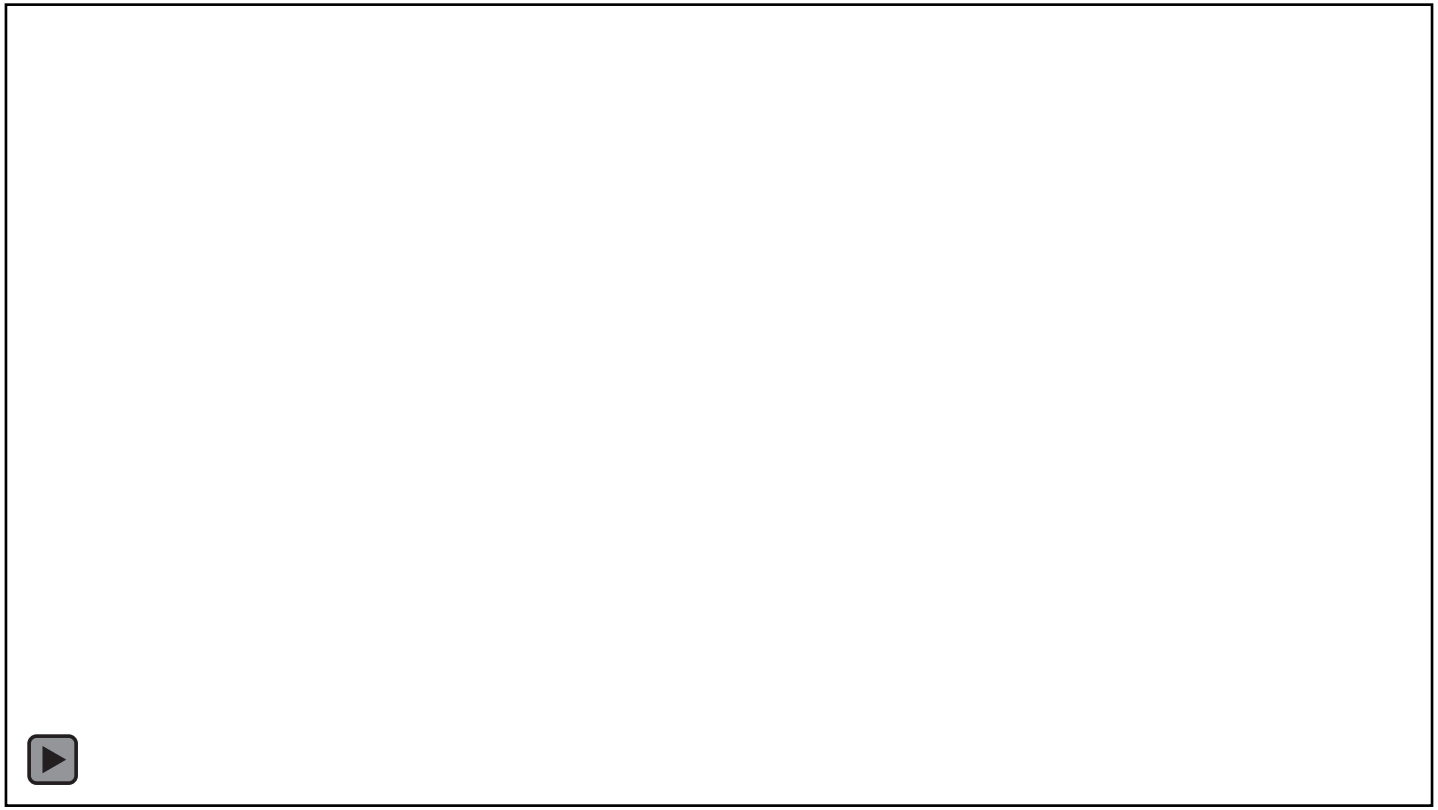
Outline

- Part I: Introduction to the AMOEBA force field
  - Physical foundations of the AMOEBA force field
  - Validation and comparison to fixed-charge force fields
- **Part II: Implementation of AMOEBA in OpenMM**
  - Accuracy and performance benchmarks
  - How to set up a simulation



Accuracy of the AMOEBA force field

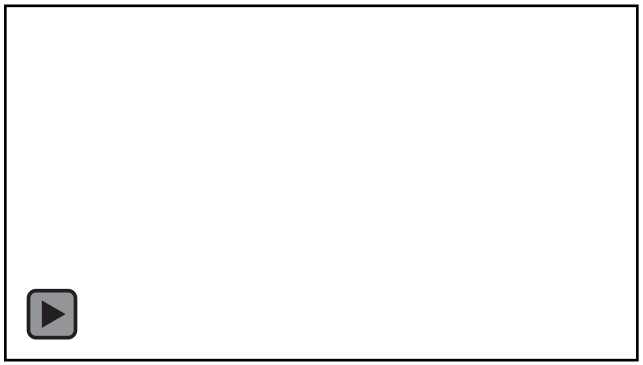
We used AMOEBA in OpenMM to compute the energies of these geometries.



A series of constrained geometry optimizations using MP2/aug-cc-pVTZ

Accuracy of the AMOEBA force field

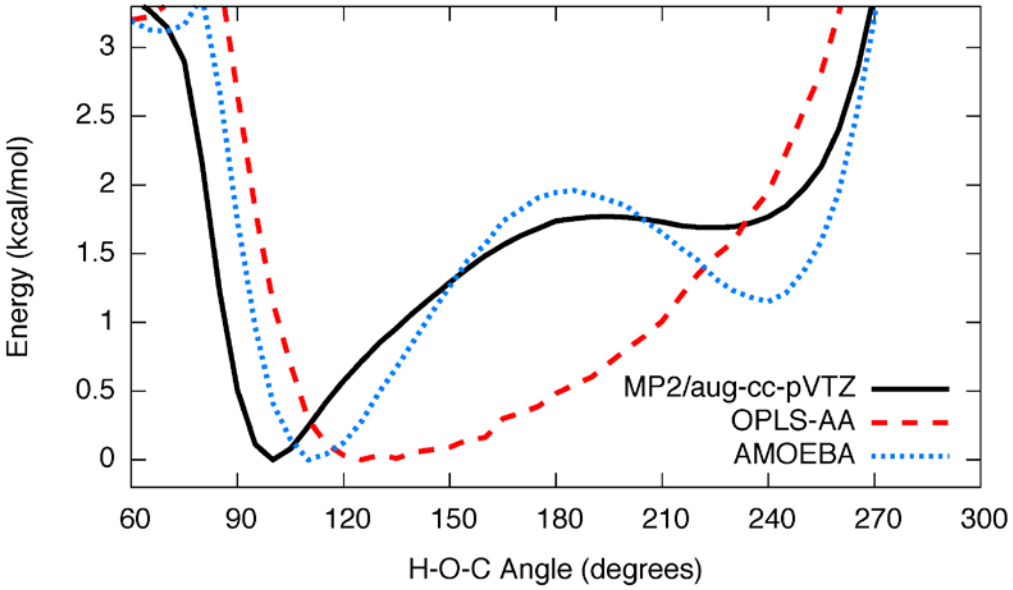
AMOEBA gives improved agreement with the *ab initio* gold standard.



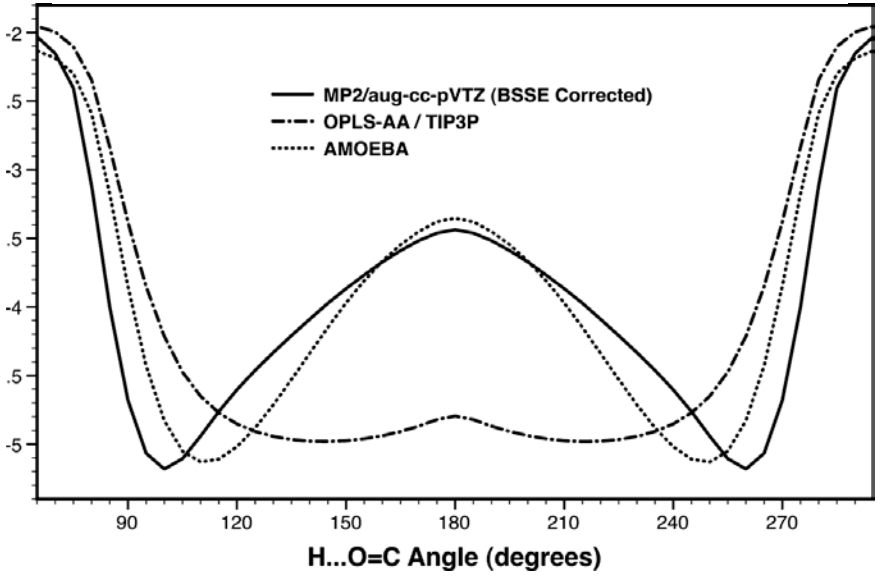
Our graph matches the left side of the literature plot and also reveals a previously hidden asymmetry!

Example available as an exercise in this Workshop

OpenMM result



Original publication



Acceleration of AMOEBA in OpenMM

Polarizable force fields have high computational cost.

Main components of increased computational cost:

- Local coordinate systems for atomic multipoles
- Iterative solution of mutually induced dipoles
- Increased complexity of bonds, angles, dihedrals and VdW terms

$$\sum_{i<j} \frac{q_i q_j}{r_{ij}}$$

Point charge electrostatics

$$\lambda_3 = 1 - \exp(-au^3)$$

$$\lambda_5 = 1 - (1 + au^3)\exp(-au^3)$$

$$\lambda_7 = 1 - \left(1 + au^3 + \frac{3}{5}a^2u^6\right)\exp(-au^3)$$

$$\lambda_9 = 1 - [1 + au^3 + (18a^2u^6 + 9a^3u^9)/35]\exp(-au^3)$$

$$\mu^{\text{ind}}_{j,\alpha}(n+1) = (1-\omega)\mu^{\text{ind}}_{j,\alpha}(n) + \omega[\mu^{\text{ind}}_{j,\alpha}(n) + \alpha_i \sum_{\{j'\}} T^{ij'}_{\alpha\beta}\mu^{\text{ind}}_{j',\beta}(n)]$$

AMOEBA electrostatics

(above equation must be solved iteratively)

$$T^{\text{D}}_{\alpha\beta} = \lambda_5 \frac{3R_\alpha R_\beta}{R^5} - \lambda_3 \frac{\delta_{\alpha\beta}}{R^3}$$

$$T^{\text{D}}_{\alpha\beta\gamma} = -\lambda_7 \frac{15R_\alpha R_\beta R_\gamma}{R^7} + \lambda_5 \frac{3(R_\alpha\delta_{\beta\gamma} + R_\beta\delta_{\alpha\gamma} + R_\gamma\delta_{\alpha\beta})}{R^5}$$

$$T^{\text{D}}_{\alpha\beta\gamma\eta} = \lambda_9 \frac{105R_\alpha R_\beta R_\gamma R_\eta}{R^9} - \lambda_7 \frac{15(R_\alpha R_\beta \delta_{\gamma\eta} + R_\alpha R_\gamma \delta_{\beta\eta} + R_a R_\eta \delta_{\beta\gamma} + R_\beta R_\gamma \delta_{\alpha\eta} + R_\beta R_\eta \delta_{\alpha\gamma} + R_\gamma R_\eta \delta_{\alpha\beta})}{R^7} + \lambda_5 \frac{3(\delta_\alpha\beta\delta_{\gamma\eta} + \delta_\alpha\gamma\delta_{\beta\eta} + \delta_{\alpha\eta}\delta_{\beta\gamma})}{R^5} \quad (4)$$

Acceleration of AMOEBA in OpenMM

OpenMM uses the GPU to evaluate the AMOEBA interactions for an enormous performance increase.

- Performance measured in ns/day with a 1 fs timestep (i.e. 1,000,000 steps)
- TINKER uses one CPU core* (Intel Core i7 @ 2.67 GHz)
- OpenMM uses one GPU (NVIDIA GeForce GTX 580)

System	Solvent	Atoms	OpenMM	TINKER	Speedup
Crambin	Implicit	642	11.4	0.053	215
Ubiquitin	Implicit	1,406	3.64	0.012	305
DHFR	Implicit	2,489	1.22	0.005	335
Water	Explicit	648	15.1	0.372	41
Villin	Explicit	3,182	2.68	0.041	65
DHFR	Explicit	23,558	0.05	0.009	52

* 2-3x speedup possible in TINKER using multiple cores, not tested here.

Using the AMOEBA force field in OpenMM

Setting up an AMOEBA simulation in OpenMM is easy and straightforward.

Try AMOEBA for the Alanine Dipeptide and Villin Headpiece examples.

- Load either **alanine-dipeptide-water.pdb** or **input_exercise3.pdb** (**input_exercise1.pdb** will not work because of residue labels)

- Make sure you're using the CUDA platform.

```
print simulation.context.getPlatform().getName()
```

- Load the AMOEBA force field (and optionally, implicit solvent) XML file

```
forcefield = ForceField('amoeba2009.xml', 'amoeba2009_gk.xml')
```

- Set the desired run parameters (note SCF convergence tolerance)

```
system = forcefield.createSystem(pdb.topology, rigidwater=False,
mutualInducedTargetEpsilon=0.0001) # Either specify PBC vectors using CRYST or keep PME
off.
```