Systematic force field optimization for more accurate molecular simulations

Lee-Ping Wang
Stanford Department of Chemistry

OpenMM Workshop, Stanford University
September 7, 2012

Outline

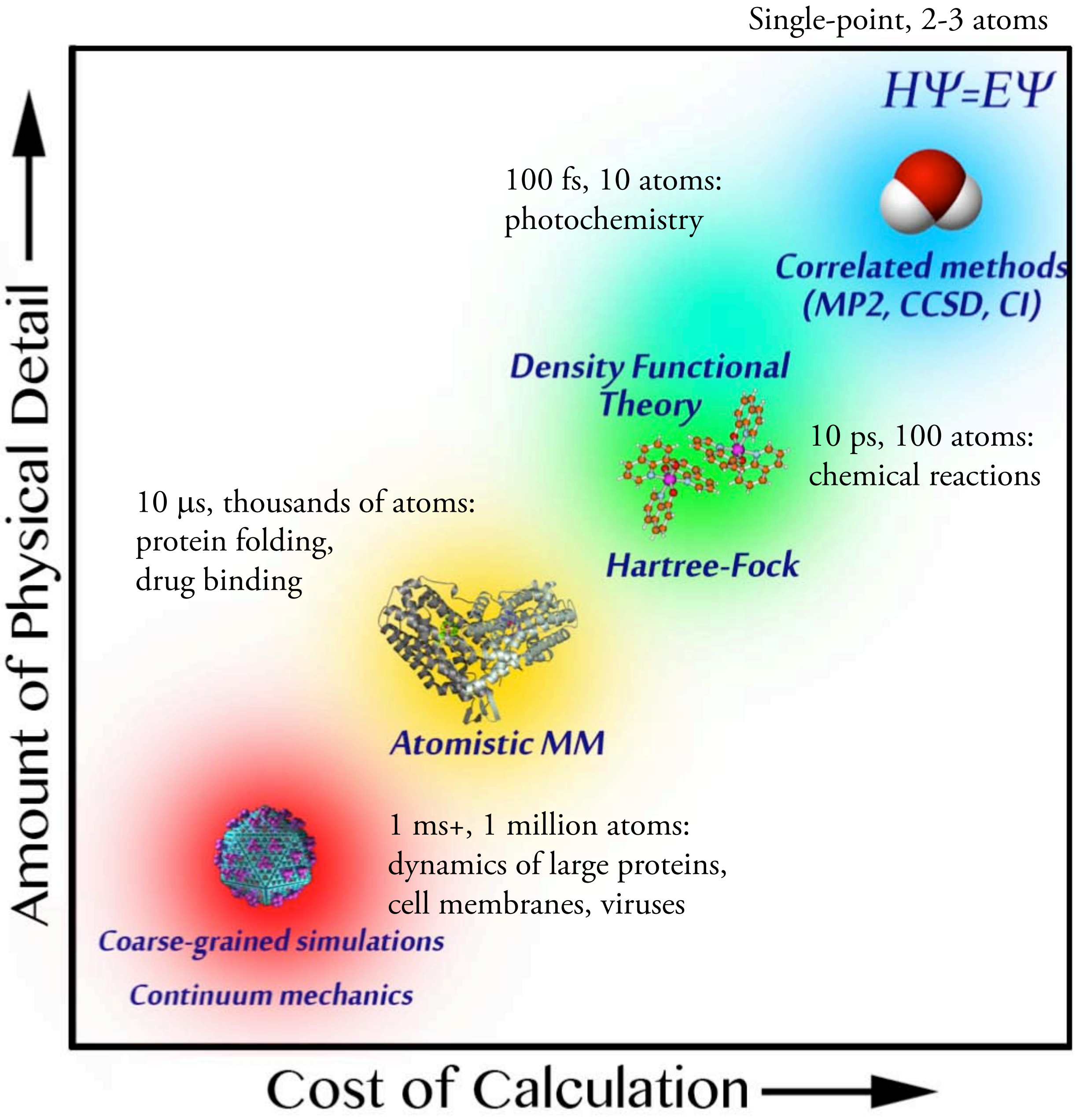
Introduction

- Force fields in molecular mechanics
- The ingredients of a force field
  - Functional form
  - Reference data
  - Optimization method

ForceBalance program for force field optimization

- Overview of program
- Application: Polarizable water model
- Results and discussion
- Basic program usage

Introduction: A wide range of simulation domains



- Computer simulations of atoms and molecules span a vast range of detail
- More detailed theories can describe complex phenomena and offer higher accuracy
- Less detailed theories allow for simulation of larger systems / longer timescales
- In molecular mechanics simulation, the potential energy of molecules is represented using an empirical force field

Introduction: Force Fields

- **Force fields** are built from *functional forms* and empirical *parameters*
- Interactions include bonded pairwise, 3-body, and 4-body interactions…
- … as well as non-bonded pairwise interactions
- Simulation accuracy depends critically on choice of parameters

Introduction: Force Fields

The common paradigm for running simulations is to choose a force field from a large literature selection.

PROTEINS:

AMBER
“Assisted Model Building with Energy Refinement”

- Main series: ff94, ff96, ff99, ff03, ff10
- Dihedral modifications: ff99sb, ff99sb-ildn, ff99sb-nmr, ff99-phi
- GAFF (Generalized AMBER force field)

OPLS
“Optimized Potential for Liquid Simulation”

- OPLS-UA (united atom), OPLS-AA (all atom)
- OPLS-AA/L (revised torsions)
- OPLS-2001, OPLS-/2005 (improved solvation free energies)

CHARMM
“Chemistry at Harvard Molecular Mechanics”

- CHARMM19 (united atom), CHARMM27 (all atom)
- CHARMM36 (carbohydrates)
- CMAP (two-dimensional dihedral corrections)
- CGenFF (General CHARMM force field)

AMOEBA
“**Atomic Multipole Optimized Energetics for Biomolecular Applications”**

- Contains polarizable point dipoles

WATER:

TIP3P, TIP4P, TIP5P
“Transferable Intermolecular Potential”

- AMBER, OPLS, and CHARMM are “paired” with TIP3P
- TIP3P water melts at -146 ºC and boils at -90 ºC

SPC, SPC/E, SPC/Fw
“Simple Point Charge”

- Same functional form as TIP3P, different parameters

TIP4P/Ew, TIP4P/Ice, TIP4P/2005

- Reparameterization of TIP4P model
- Improved fits to experimental properties of water

Various polarizable models

- SWM4-DP, SWM4-NDP (contains Drude particle)
- AMOEBA (contains polarizable point dipoles)
- DPP, DPP2 (distributed point polarizable model)
- TTM2-F, TTM2-R, TTM3-F (Thole type model)
- TIP4P-FQ, SPC-FQ (Fluctuating charge model)

There are too many to choose from…

Can we create a force field that is best for our research project?

Creating a force field: Functional form

Step 1: Choose a functional form to represent the potential energy surface, or design your own.

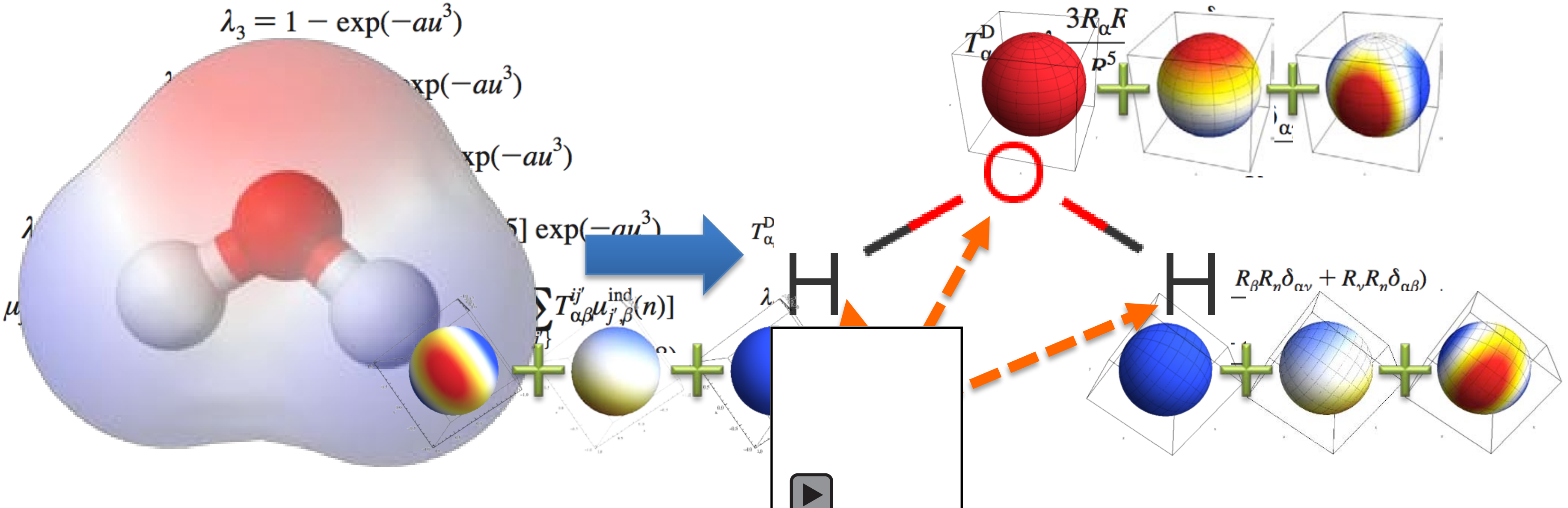
AMBER fixed-charge force field:

- Point charge on each atom

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

AMOEBA polarizable force field:

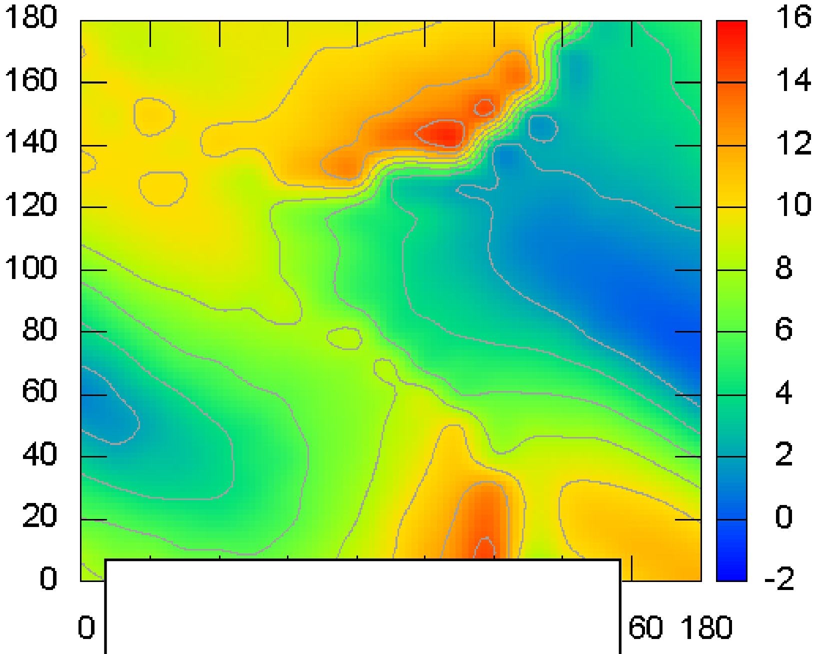
- Point charge, dipole, and quadrupole on each atom
- Polarizable point dipole on each atom with short-range damping


$\lambda_3 = 1 - \exp(-au^3)$
$\mathrm{xp}(-au^3)$
$\mathrm{xp}(-au^3)$
$5]\exp(-au^3)$
$\sum T^{ij'}_{\alpha\beta} u^{\mathrm{ind}}_{j',\beta}(n)]$
T^{D}_{α}
$3R_\alpha R$
R^5
O
H
H
$R_\beta R_n \delta_{\alpha\nu} + R_\nu R_n \delta_{\alpha\beta})$

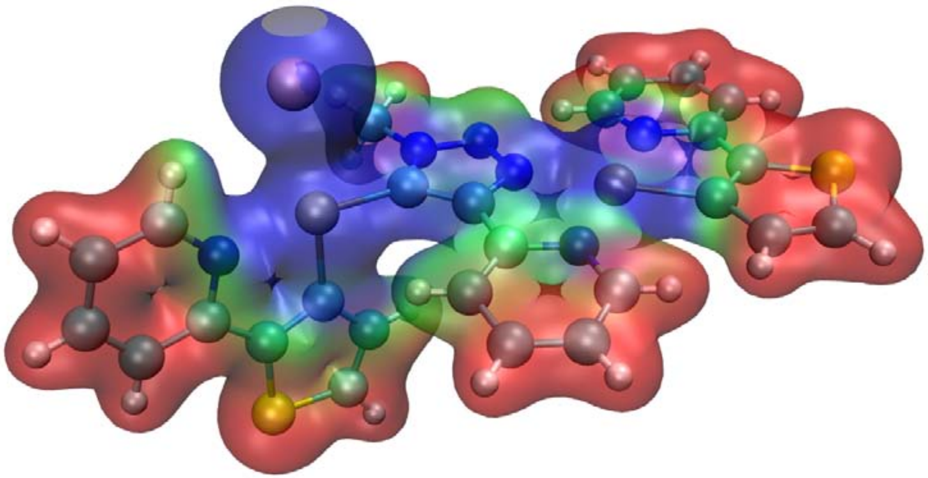
Creating a force field: Reference data

Step 2: Create a reference data set from theoretical calculations or experimental measurements.

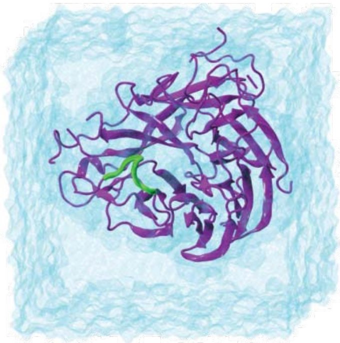
Energy scan across 2 dihedral angles

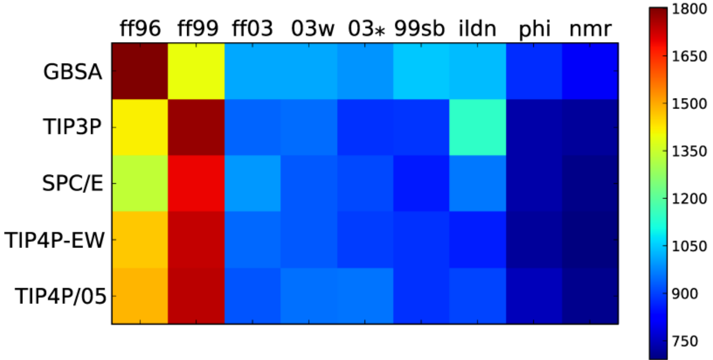


Electrostatic potential on a molecular surface (red = positive, blue = negative)



Simulated vs. experimental NMR chemical shifts for proteins (red = bad, blue = good)





Creating a force field: Optimization method

Step 3: Construct an objective function and apply an optimization method to minimize it.

- The **objective function** measures the disagreement between the reference data and corresponding simulation result.

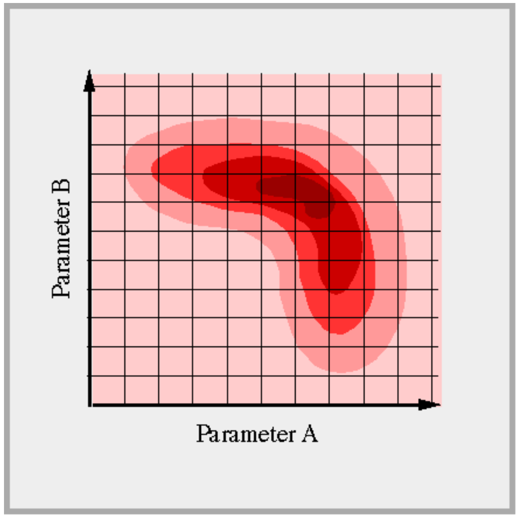
- An **optimization algorithm** searches for parameters that minimize the objective function.

R = Reference Data

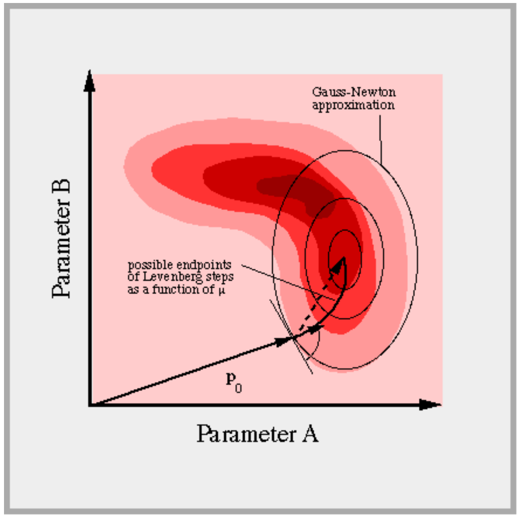
S = Simulation Result

$$\chi^2(\mathbf{k}) = \left(R - S(\mathbf{k})\right)^2$$

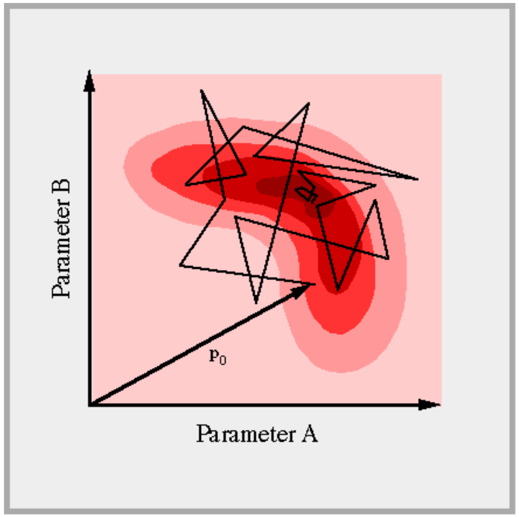
$$\mathbf{k}_{opt} = \min_{\mathbf{k}} \chi^2(\mathbf{k})$$



Grid Scan


Gauss-Newton approximation
possible endpoints of Levenberg steps as a function of μ
P0
Parameter B
Parameter A

Newton-Raphson



Simulated Annealing

Outline

Introduction

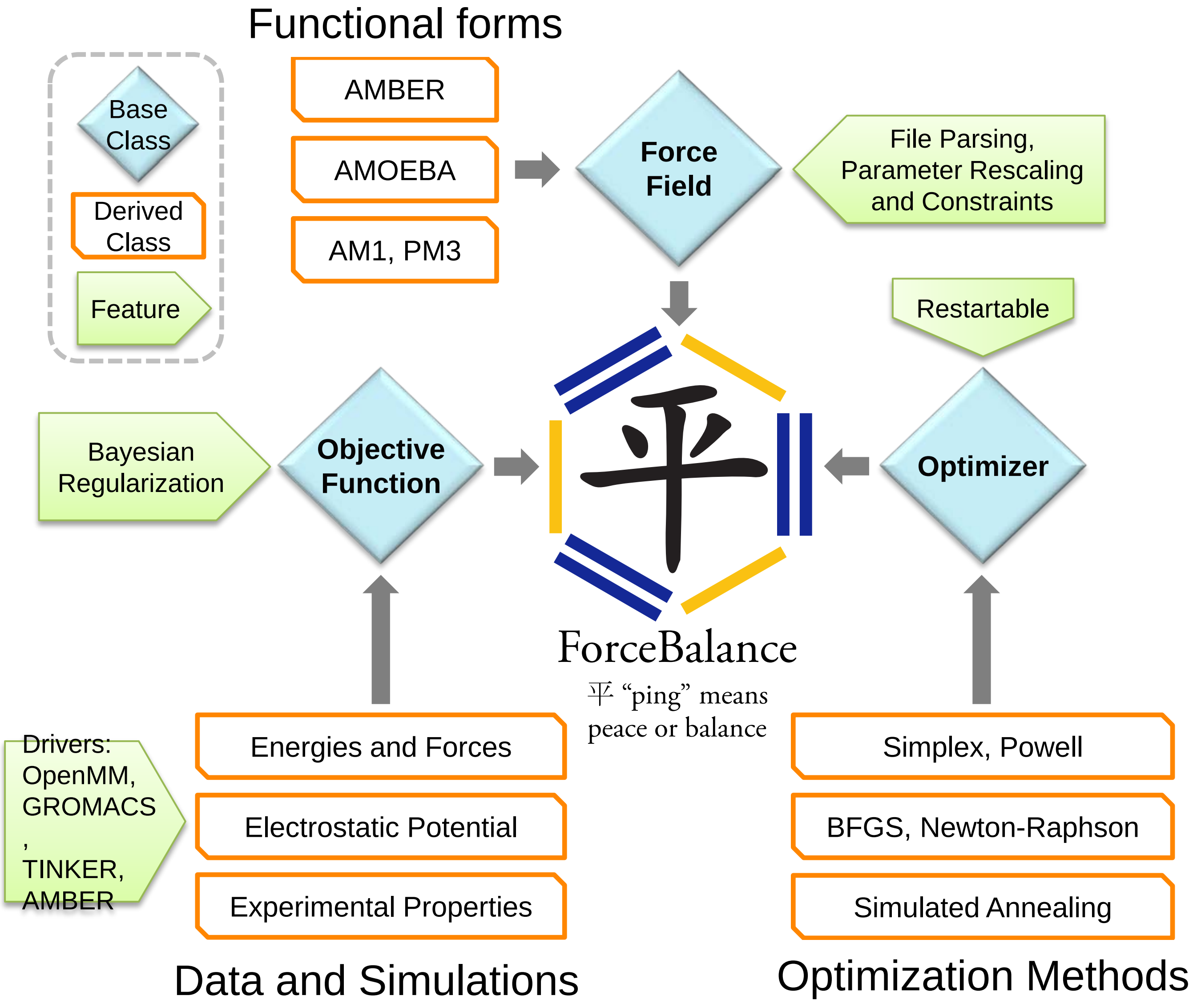
- Force fields in molecular mechanics
- The ingredients of a force field
  - Functional form
  - Reference data
  - Optimization method

ForceBalance program for force field optimization

- Overview of program
- Application: Polarizable water model
- Results and discussion
- Basic program usage

Introducing ForceBalance

ForceBalance is free software for creating force fields.



Functional forms

Base Class

Derived Class

Feature

AMBER

AMOEBA

AM1, PM3

Force Field

File Parsing, Parameter Rescaling and Constraints

Restartable

Bayesian Regularization

Objective Function

平

Optimizer

ForceBalance

平 “ping” means peace or balance

Drivers: OpenMM, GROMACS , TINKER, AMBER

Energies and Forces

Electrostatic Potential

Experimental Properties

Simplex, Powell

BFGS, Newton-Raphson

Simulated Annealing

Data and Simulations

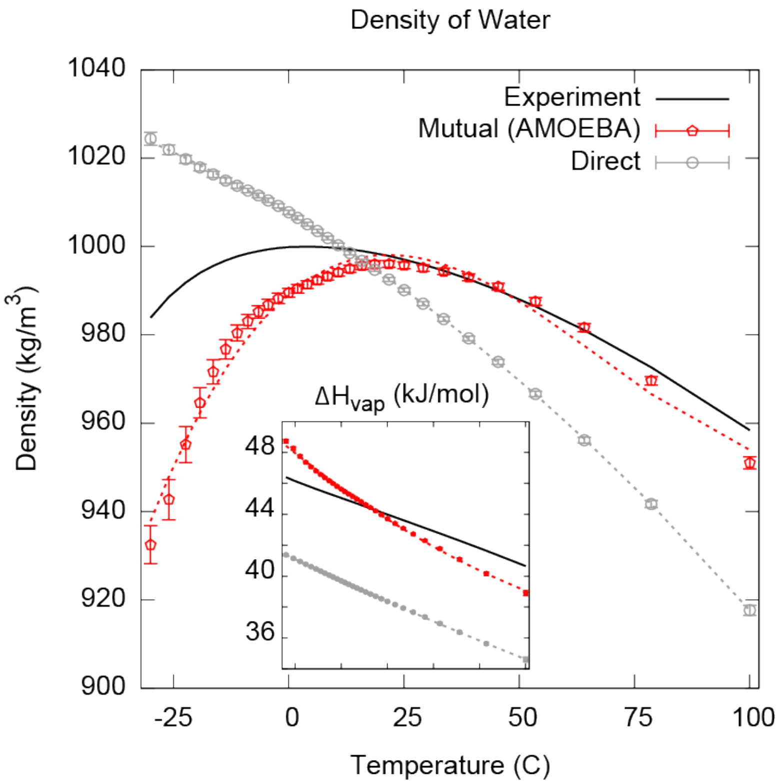
Optimization Methods

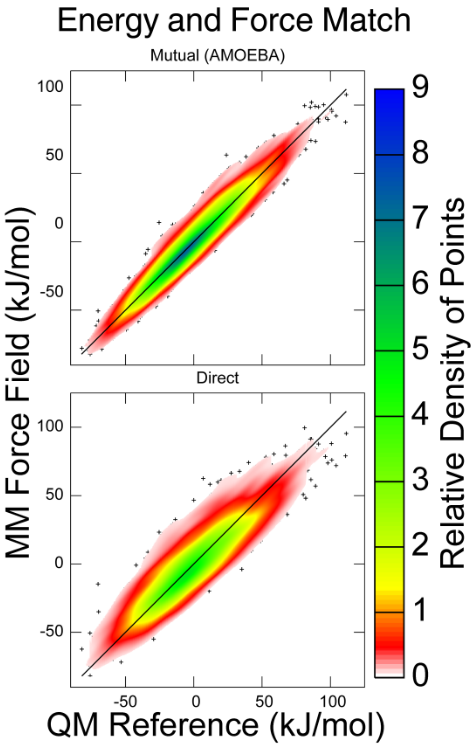
- Written in Python
- Direct interface with OpenMM
- Highly flexible and easily extensible
- Freely available at simtk.org with installation instructions and user’s manual

http://simtk.org/home/forcebalance

Polarizable water - motivation

We applied ForceBalance to parameterize a variation on the AMOEBA water model.

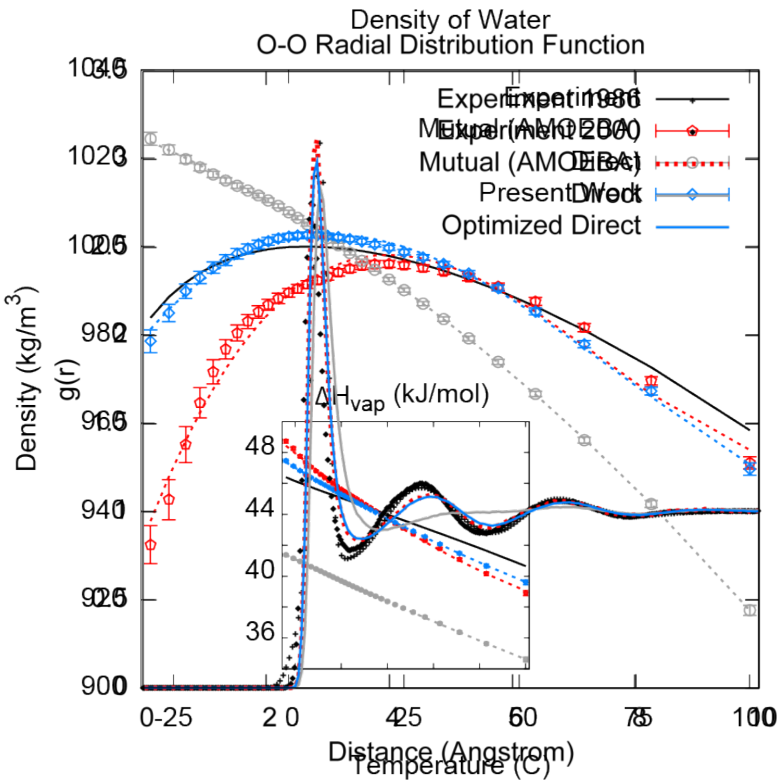


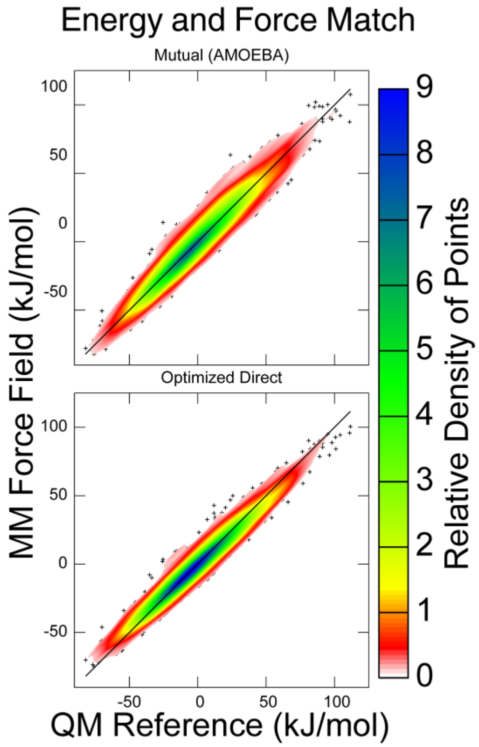


- The AMOEBA force field contains mutually induced dipoles
- Direct induced dipoles are cheaper (5x faster) but the physics of the model are different
- 19 total tunable parameters

Polarizable water - results

Our optimized model exceeds the accuracy of AMOEBA for several properties of water.





Property	AMOEBA	This work	Experiment
Density (kg m^{-3})	1000 ± 1	999 ± 1	997
ΔH_{vap} (kJ mol^{-1})	43.8 ± 0.1	43.8 ± 0.1	44.0
Dielectric constant	81 ± 10	81 ± 5	78.4
Diffusion constant (10^{-5} cm^2 s^{-1})	2.0 ± 0.1	2.3 ± 0.1	2.3
Density maximum (ºC)	15 - 25	0 - 10	4

- We used a large set of experimental and theoretical data:

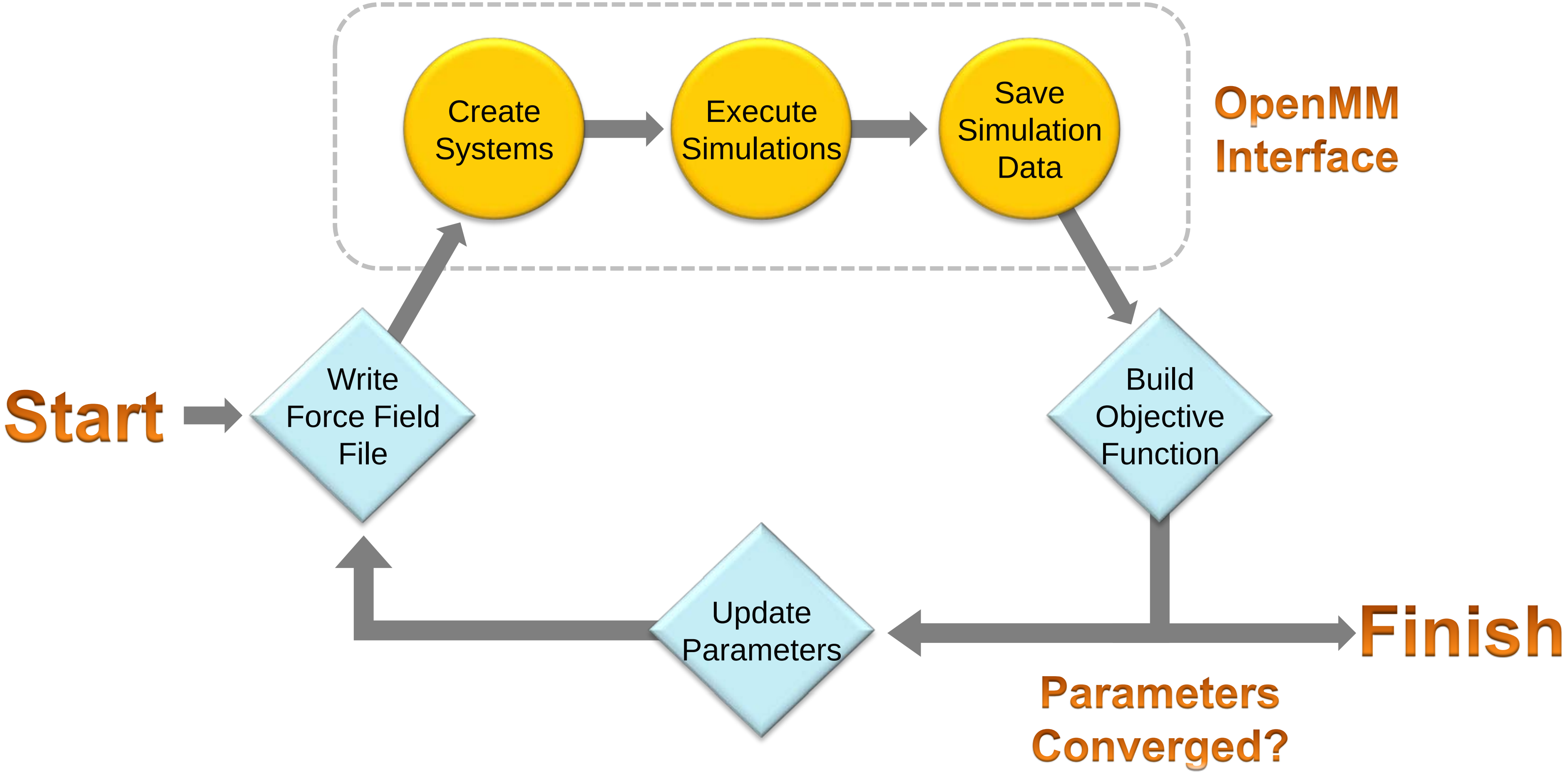
1) Energies and forces for 12,000 geometries from QM theory
2) Gas-phase cluster binding energies from QM theory
3) Experimental monomer geometry, vibrational modes, and multipole moments
4) Experimental density and heat of vaporization curves

- Fitted properties exceed accuracy of original AMOEBA

- Other properties were also predicted accurately

Interface to OpenMM

ForceBalance interfaces with OpenMM by importing it as a Python module.



Force field parameter files

The parameters to be optimized are specified by labeling the XML file.

- Simply add a “parameterize” attribute to the XML element containing parameters to be optimized.

- At each optimization step, ForceBalance writes new parameter files containing updated parameter values.

- Several other force field formats are supported (GROMACS `.itp`, AMBER `.mol2` and `.frcmod`, TINKER `.prm`

- Parameters can either be independent variables or arbitrary functions of other parameters (advanced functionality).

Example of labeled force field XML file

```
<AmoebaHarmonicBondForce>
  <Bond class1="73" class2="74" length="0.09572" k="232986.04" parameterize="length, k" />
</AmoebaHarmonicBondForce>
 <AmoebaHarmonicAngleForce>
  <Angle class1="74" class2="73" class3="74" k="0.06207" angle1="108.50" parameterize="angle1, k" />
</AmoebaHarmonicAngleForce>
```

ForceBalance input file

The optimization is completely specified using the input file.

- Generate a documented input file with all available options with `MakeInputFile.py`
- Set up directories containing reference data and simulation settings
- Run the optimization using `ForceBalance.py`
- Optimizations can be restarted by pasting sections of output back into input

Example of ForceBalance input file

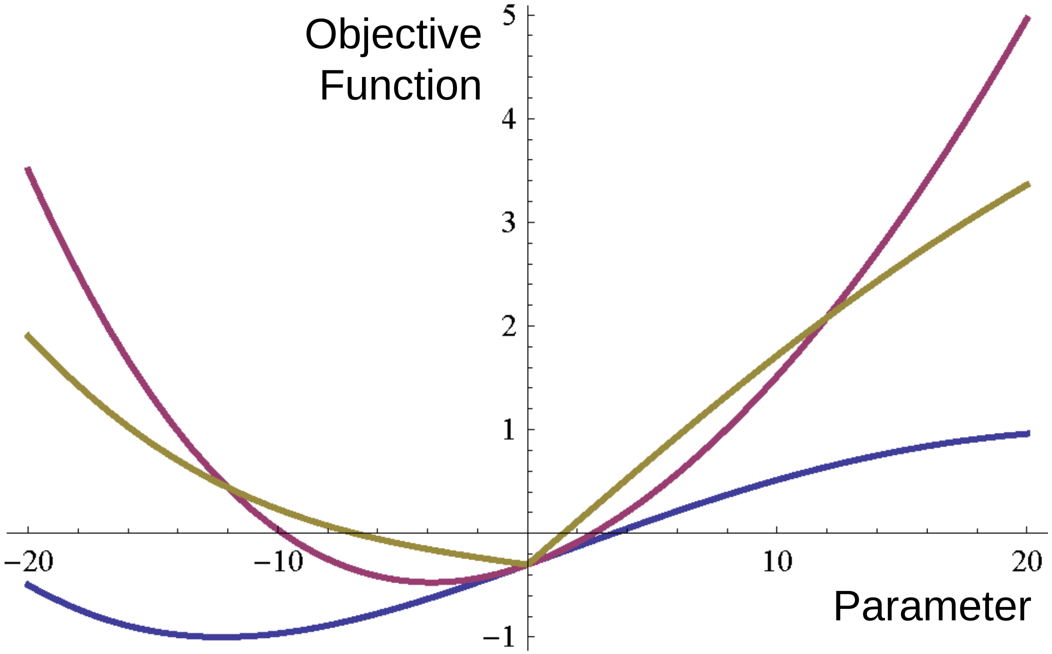
```
$options
jobtype    newton
tinkerpath /home/leeping/opt/tinker-6.1.01-intel/bin/
forcefield amoebawater.xml water.prm
trust0     0.1 # Levenberg-Marquardt trust radius
$end

$simulation
name       LiquidCluster-12
simtype    AbInitio_OpenMM
$end
```

Bayesian regularization

Optimizations with hundreds of parameters are made possible through strict regularization.

- We address overfitting issues by applying a Bayesian *prior*.
- The prior affects the optimization by penalizing large parameter movements.
- Different types of priors (Gaussian, Laplacian) have various impacts on the optimization behavior



Gaussian prior (L2 regularized):
Large movements penalized

Laplacian prior (L1 regularized):
Some parameters don’t change

No regularization:
Prone to overfitting

Conclusion

We hope that ForceBalance will *systematize* and *democratize* the discipline of force field development.

Systematic optimization methods:

- Optimize parameters using theoretical and experimental data simultaneously
- Parameterization calculations are reproducible and systematically improvable
- Rigorously prevent overfitting using strict regularization methods

Give everybody the infrastructure for making good force fields:

- Improve simulation accuracy for uncommon (non-mainstream) molecules, where force field development efforts are relatively sparse
- All-inclusive: New interfaces with simulation software are easy to write
- Reduce the headache of force field development and let's focus on the science

http://simtk.org/home/forcebalance