**Poisson Boltzmann – Analytical Model (PB-AM)**

**and**

**Poisson-Boltzmann Semi-analytical Model (PB-SAM)**



**REFERENCE MANUAL**

Principal contributors to the current codes:

Enghui-Yap (Albert Einstein College of Medicine)

Lisa Felberg (University of California, Berkeley)

Teresa Head-Gordon (University of California, Berkeley)

For more information, please visit <http://thglab.berkeley.edu>

**Acknowledgments:** Research support from NIH and DOE is gratefully acknowledged.

**Cover Illustration:** An exploded view of a Brome Mosaic Virus capsid composed of T = 1 particles (PDB: 1YC6), represented as collections of overlapping spheres, is shown. PB-SAM is a new semi-analytical approach to efficiently solve the linearized Poisson—Boltzmann equation using multipole formalisms for overlapping spheres. The background shows the potential profile for an array of 601YC6 monomers computed using this method.

**Recommended Citations:** When citing PB-SAM in the literature, the following citations should be used

1. O. N. Demerdash, E.-H. Yap and T. Head-Gordon (2014). Advanced potential energy surfaces for condensed phase simulation. *Ann. Rev. Phys. Chem.* 65*,* 149-174
2. E.-H. Yap and T. Head-Gordon (2013). Calculating the bimolecular rate of protein–protein association with interacting crowders. *J. Chem. Theory Comput.* 9(5), 2481-2489.
3. E.-H. Yap and T. Head-Gordon (2010). New and efficient Poisson-Boltzmann solver for interaction of multiple proteins  *J. Chem. Theory Comput. (Journal cover)* 6, 2214-2224.

When citing PB-AM in the literature, the following citations should be used

1. I. Lotan & T. Head-Gordon (2006). An analytical electrostatic model for salt screened interactions between multiple proteins *J. Chem. Theory Comput* 2, 541-555.

Table of Contents

Introduction 3

Installation 4

PB-AM Installation 4

PB-SAM Installation 4

MC-PB-SAM Installation 4

PB-AM Basics 5

File usage 6

Example input files and input file information 6

Program tutorials 7

Two versus three body interactions – cng flag 7

Potential of multi-bodies in solution – dif/rad flag 7

Infinte grid calculation – inf flag 8

Polarization – pol flag 8

BD simulation – sim flag 9

Energy calculation – slv flag 9

PB-SAM Basics 10

File usage 10

Program tutorials 10

**Introduction** 1

Installation of PB-AM and PB-SAM 2

OpenMP Parallelization

**PB-AM Basics**

Electrostatic Potentials

**PB-SAM Basics**

Biomolecular Rates using Northrup-Allison-MacCammon

Biomolecular Rates using PBC and MFPT

**Monte Carlo Rendering of Initial Configurations for PB-SAM**

File usage 3

Example input files and input file information 4

Suggested parameter settings 5

Bibliography

# Introduction

PB-AM and PB-SAM describe two different methods and an associated suite of programs that allow users to carry out linearized Poisson-Boltzmann Equation solutions to systems hitherto not possible using traditional PBE solvers. The PBE software suite is licensed as a collection of freely available program under a GPL license, and is divided into three parts:

**PB-AM** is the first general analytical solution for computing the screened electrostatic interaction between *large numbers* of macromolecules of arbitrarily complex charge distributions, assuming they are well described by spherical low dielectric cavities in a higher dielectric medium in the presence of a Debye-Hückel treatment of salt. The method exploits multipole expansion theory for the screened Coulomb potential such that it can describe direct charge-charge interactions and *all* higher-order cavity polarization effects between low dielectric spherical cavities containing their charges, while treating these higher order terms correctly at all separation distances. The analytical solution is general to arbitrary numbers of macromolecules, is efficient to compute, provides for the first time the ability to provide new benchmarks for other numerical solutions to the linearized Poisson-Boltzmann equation.

**PB-SAM** is a new numerical approach to solving the linearized Poisson Boltzmann equation by representing the macromolecule surface as a collection of spheres in which the surface charges can then be iteratively solved by the PB-AM analytical multipole method. Our Poisson Boltzmann semi-analytical method, PB-SAM, realizes better accuracy, more flexible memory management, and at reduced cost relative to either finite difference or boundary element method PBE solvers. In addition, we have extend the applicability of the PB-SAM approach by deriving force and torque expressions that fully account for mutual polarization in both the zero and first order derivative of the surface charges, that we have embedded into a Brownian dynamics scheme to look at electrostatic-driven mesoscale assembly and kinetics. This allows for the simulation of protein concentration effects and crowding conditions on the biomolecular rate of under wither a Northrup-Allison-McCammon approach in addition to our new formulation of rates using periodic boundary conditions and evaluated through mean first passage times.

**MC-PB-SAM.** The PB-SAM approach renders the molecular surface as a collection of overlapping spheres whose resolution is controlled by the sphere size used. The solvent excluded molecular surface (SES) is determined and the resulting SES and PQR files are inputs into MC-PB-SAM program to discretized the macromolecule(s) into spheres. At each iteration, this program uses a greedy Monte Carlo algorithm to search for a sphere center that encompasses the largest number of fixed partial charges. Charges within this sphere center are then removed, and the search is repeated with the remaining charges. A stipulated tolerance controls how far from the SES the sphere surface can protrude.

## Installation

### PB-AM Installation

To install PB-AM from the source code, first pull the latest version from the github/bitbucket site and type the following into the command line:

make mpe

This should make the executable, *mpe*, and place it in the *bin* directory of the source code

### PB-SAM Installation

### MC-PB-SAM Installation

# PB-AM Basics

PB-AM is an analytical solution to the linearized Poisson-Boltzmann equations for multiple spherical objects of arbitrary charge distribution in an ionic solution. The linearized Poisson-Boltzmann equation is given as:

Exploiting fast-multipole methods, this boundary value problem can be reduced to the following system of linear equations.

**A(i)** represents the effective multipole expansion of the charge distributions of molecule (i). **E(i)** is the free charge distribution of molecule (i). Γ is a dielectric boundary-crossing operator, Δ a cavity polarization operator, T an operator that transforms the multipole expansion to a local coordinate frame. More details on the method are available in Lotan, Head-Gordon (2006).

From this formulation, computation of the interaction energies (Ω**(i)**) is given as follows:

Where < M, N > denotes the inner product. When energy is computed, forces follow as:

## File usage

From the single *mpe* executable, multiple types of calculations can be performed. This section will cover each calculation, its required inputs and it’s outputs.

## Example input files and input file information

Generally, all the programs require a computation flag, and a PDB or PQR file name.

If a PDB file is chosen the input is read and atoms are assigned partial charges according to the file *charges\_OPLS,* located in each of the test directories.

A PQR file can be generated from the online site: <http://nbcr-222.ucsd.edu/pdb2pqr_1.9.0/> or in the downloadable executable from <http://www.poissonboltzmann.org/docs/pdb2pqr-installation/> . It may also be formatted manually.

The general format of a PQR file is as follows, and is whitespace-delimited:

recName serial atName resName chainID resNum X Y Z charge rad

*recName* A string that should either be ATOM or HETATM.

*serial* An integer that provides the atom index

*atName* A string that provides the atom name.

*resName* A string that provides the residue name.

*chainID* An optional string that provides the chain ID of the atom.

*residueNumber* An integer that provides the residue index.

*X Y Z* Three floats that provide the atomic coordinates.

*charge* A float that provides the atomic charge (in electrons).

*Rad* A float that provides the atomic radius (in Ã…).

## Program tutorials

### Two versus three body interactions – cng flag

Executes a program that computes the effect of a third molecule on the interaction of free energy of two molecules placed equidistantly. It requires the following input:

./exec cng [PQR file] [distance between molecles] [runname]

The following files should be in the CWD: \*pqr and charges\_OPLS

And it outputs the following:

[runname]\_dist.txt

This file has 900 lines, each line representing the interaction free energy between two molecules. The first value in each line represents the system of just two molecules, the next 30 values represents including a 3rd molecule in the system at 30 different orientations.

### Potential of multi-bodies in solution – dif/rad flag

dif\_test

Executes the computation of the potential at points on a grid in space from -75 to +75, for a system of 2, 4, 6, or 8 molecules placed equidistantly in a salty solution. It requires the following input:

./exec dif [PQR file] [2, 4, 6 or 8 molecules] [distance between molecules]

The following files should be in the CWD: \*pqr and charges\_OPLS

And it outputs the following:

pqrname\_[# part]P.mdp

A PQR file of file of molecules in their positions created by dif run with original OPLS partial charges

pqrname\_[# part]P\_sp.mdp

Same as first PQR file, but with CG spheres included as well

pqrname\_[# part]P.gmpe

A file of gridded potentials at 151 points for the X, Y and Z dimension.

-------------------------

Executes the self-polarizarion and potential calculation on a grid for a single CG molecule, positioned at (0,0,0). Similar to dif option, but for only one molecule. It requires the following input:

./exec rad [PQR file] [scaling factor]

The following files should be in the CWD: \*pqr and charges\_OPLS

And it outputs the following:

pqrname\_[# part]P.mdp

A PQR file of file of molecules in their positions created by dif run with

original OPLS partial charges

pqrname\_[# part]P\_sp.mdp

Same as first PQR file, but with CG spheres included as well

pqrname\_[# part]P.gmpe

A file of gridded potentials at 151 points for the X, Y and Z dimension.

### Infinte grid calculation – inf flag

Executes the creation of an infinite grid approximation. It requires the following input:

./exec inf [PQR file] [# lattice layers] [stretch factor]

The following files should be in the CWD: \*pqr and charges\_OPLS

And it outputs the following:

pqrname\_88.pdb

A PDB file of the input molecule with [# lat lay] layers in a grid

\*.out

The program prints out the potential, the XYZ force and the XYZ torque

for the eight molecules placed at the center of the infinite grid

### Polarization – pol flag

Executes a routine the compares the force and torque of identical molecules places equidistantly in a salty environment, with and without mutual polarization effects. It requires the following input:

./exec pol [PQR file] [2, 4, 6 or 8 molecules] [distance between molecules] [runname]

The following files should be in the CWD: \*pqr and charges\_OPLS

And it outputs the following:

polar\_[force/torque]\_name\_nmol\_dist.txt

The first line indicates the force or torque computed per molecule in the absence of mutual polarization. The next line includes mutual polarization. These 2 lines repeat 1000 times.

### BD simulation – sim flag

Executes the BD simulation part of the code. It requires the following input:

./exec sim [Salt conc] [outfile] [temp file #]

The following files should be in the CWD: \*pqr and charges\_OPLS

And it outputs the following:

temp[temp file #].out

A temp file that outputs parameters while running each BD run

[outfile]

A File that contains the outcome of each BD run. the first

column is the status of the simulation at the end, the second

is the trajectory number and the last column is the step number

### Energy calculation – slv flag

Computes the energies, forces and the torques on a system of multiple molecules, placed equidistantly in a salty solution. It requires the following input:

./exec slv [PQR file] [2, 4, 6 or 8 molecules] [distance between molecules]

The following files should be in the CWD: \*pqr and charges\_OPLS

And it outputs the following:

\*.out

A list of each molecule in the system and the energy, force and torque acting on it

# PB-SAM Basics

## File usage

## Program tutorials