

User Guide (Binary Version 0.1, December 9, 2017)

Regularized Matched Interface & Boundary Poisson-Boltzmann solver (rMIB)

1. Introduction

This guide gives direction to use the binary Regularized Matched Interface and Boundary Poisson-Boltzmann solver (rMIB) [1]. The rMIB PB solver is a suitable choice if the accuracy of electrostatic potentials and fields are the major concerns of one's research. The implementation requires a triangular mesh to represent the molecular surface, which is generated by MSMS [2].

2. System Requirements

Users can download the binary file for their running systems. Currently, the binary files for Linux and are available. One needs to download the following files suitable for their systems.

(1) rMIB.exe

(2) msms (saved as “msms” for Linux and Mac)

(3) usrdata.in

(4) test_proteins (a folder with test proteins in forms of APBS's .pqr file, users can go to the web portal “<http://pdb2pqr.ncr.net>” to generate the .pqr file for interested biomolecules [3])

Alternatively, one can put msms elsewhere and included it in system path (type “msms” to test). Users can download and install MSMS from “<http://mgltools.scripps.edu/packages/MSMS/>”.

3. Input

The input parameters are set in the file “usrdata.in”. The following is a sample file with explanations.

```
fname 1ajj           #PDB ID for testing proteins
den 10              #MSMS density (vertices per angstrom^2)
epsf 1             #dielectric constant for protein
epsw 80            #dielectric constant for solvent
bulk_strength 0.15 #ionic concentration (M)
icg 3              #0:no charge; 1:linear interp.; 2:Green's func.
isf 1              #0:exact surface; 1: msms interface;
ibd 3              #0:exact soln; 1: Kirkwood; 2: 1/r; 3:exp
irg 1              #0:3-term (mib[5]); 2:2-term (rmib[1])
inl 0              #0: linear PB; 1: nonlinear PB
```

Type “./rMIB.exe” to run TABI solver. Make sure “usrdata.in” is in the same directory as rMIB.exe, and your target proteins in the format of APBS's .pqr files are stored in the subdirectory called “test_proteins”.

4. Output

- 1) The rMIB code produces an output file called pot.dat containing surface potentials [$k_b T/e_c$].
- 2) The rMIB code also prints the free energy of solvation [kcal/mol] on the screen together with some other information such as CPU time and the BiCG residuals at each step.

5. Application

1) rMIB can be used to calculate pKa (co-logarithm of the acid dissociate constants) for titrating sites of solvated proteins [3].

6. Reference

- [1] W. Geng and S. Zhao, *A two-component Matched Interface and Boundary (MIB) regularization for charge singularity in implicit solvation*, J. Comput. Phys., 351, 25-39 (2017).
- [2] M.F. Sanner, A.J. Olson and J.C. Spehner, *Reduced surface: An efficient way to compute molecular surfaces*, Biopolymers, 38, 305-320 (1996).
- [3] N.A. Baker, D. Sept, S. Joseph, M.J. Holst, J.A. McCammon, *Electrostatics of nanosystems: application to microtubules and the ribosome*. Proc. Natl. Acad. Sci. USA, 98, 10037-10041 (2001).
- [4] J. Hu, S. Zhao, W. Geng, *Accurate pKa computation using matched interface and boundary (MIB) method based Poisson-Boltzmann solver*, Commun. Comput. Phys., 32, 520-539 (2017).
- [5] W. Geng, S. Yu and G.W. Wei, *Treatment of charge singularities in implicit solvent models*, J. Chem. Phys., 127, 114106-114126 (2007).

7. Acknowledgment

NSF Grant DMS-0915057, NSF Grant DMS-1318898, DMS-1418957.

Notes: For users who received source code

- Source Term (icg)

icg=0: for the situation that the RHS of Poisson-Boltzmann equation is a given analytical function $f(x)$; (the function is given in para_func.oneb.f90, and the user can use this option only after fully understanding of the source code)

icg=1: the singular charge on the RHS of the PB equation is distributed to the neighboring grids by trilinear interpolation; this is essentially the MIBPB-I,II implementation [5];

icg=2: use Green's function based regularity; MIBPB-III [5] and rMIB [1]

For source code:

Check functions in chgdist.90, SetChg() in mibpb_para.f90, and setChgPos() in topology.f90

- Boundary Condition (ibd)

The solution at the cut-off edge of the Cube as the finite difference domain

ibd = 0: closed-form analytic solution for a sphere with centered charge;

ibd = 1; spherical-expansion solution for a sphere with distributed charges;

ibd = 2; molecular surface other than sphere with $1/r$ decaying for Poisson equation;

ibd = 3; molecular surface other than sphere with $\exp(-kr)/r$ decaying for PB equation;

When option 0 and 1 are chosen, the code returns surface potential errors.

For source code:

Check Setb() in main.f90, which calls bdcond() in mibpb_para.f90 and some related functions.

- Interface Conditions (isf)

isf = 0: exact interface, which essentially the sphere. Two-sphere system is available too but needs to replace para_interface.f90 by para_interface_twob_mov.f90 (for dynamics);

isf = 1: molecular surface generated by MSMS.

- Regularity (irg)

irg = 0: three-term Green's function [5];

irg = 1: two-term Green's function [1]

- Nonlinearity (inl)

inl = 0: linear PB

inl = 1: nonlinear PB

For source code:

The nonlinear code is in nonlinearsolver.f90.