

# Lecture 5: Numerical Methods for Biomolecular Simulations

Guowei Wei

Mathematics

Michigan State University

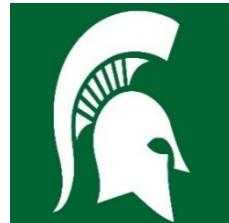
<https://users.math.msu.edu/users/wei/>

NSF-CBMS Conference on Mathematical Molecular Bioscience and Biophysics

University of Alabama

Tuscaloosa, May, 13-17, 2019

**Grant support: NSF, NIH, MSU, BMS, and Pfizer**



# Implicit Solvent Models— Poisson-Boltzmann

Electrostatic free energy functional  $G$  (Sharp and Honig, 1990)

$$G = \int [\text{Polar}] d\mathbf{r}$$

Polar = electric field + solute charges + solevent charges:

$$\text{Polar} = \phi \rho_m - \frac{\varepsilon}{2} |\nabla \phi|^2 - kT \sum_j c_j \left( e^{-\frac{q_j \phi}{kT}} - 1 \right)$$

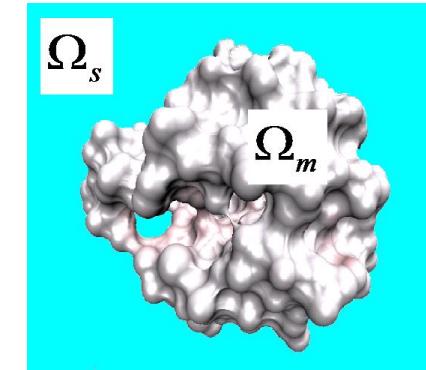
$$\varepsilon(\mathbf{r}) = \begin{cases} \varepsilon_m & \forall \mathbf{r} \in \Omega_m \\ \varepsilon_s & \forall \mathbf{r} \in \Omega_s \end{cases}$$

$$\rho_m = \sum_i (Q_i \delta(\mathbf{r} - \mathbf{r}_i) - \cancel{d_i \cdot \nabla \delta(\mathbf{r} - \mathbf{r}_i)} + \cancel{\Theta_i \cdot \nabla \nabla \delta(\mathbf{r} - \mathbf{r}_i)})$$

$$\text{Euler-Lagrange equation: } \frac{\partial g}{\partial \phi} - \frac{\partial}{\partial \mathbf{r}} \frac{\partial g}{\partial \phi} = 0$$

Poisson-Boltzmann equation:

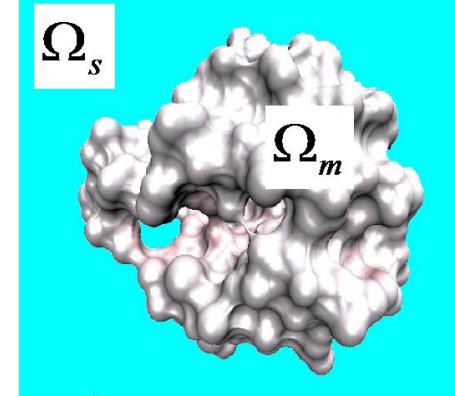
$$-\nabla \cdot (\varepsilon(\mathbf{r}) \nabla \phi) - \sum_j q_j c_j e^{-\frac{q_j \phi}{kT}} = \rho_m$$



# Interface and boundary conditions of the Poisson Boltzmann equation

The Poisson-Boltzmann equation

$$-\nabla \cdot (\varepsilon(\mathbf{r}) \nabla \phi) - \sum_j q_j c_j e^{-\frac{q_j \phi}{kT}} = \rho_m$$



For  $\phi$  and  $\varepsilon(\mathbf{r}) \nabla \phi$  to be differentiable at the interface

$$\begin{cases} \phi^- = \phi^+ & \forall \mathbf{r} \in \partial\Omega_m \\ \varepsilon_m \frac{\partial \phi^-}{\partial n} = \varepsilon_s \frac{\partial \phi^+}{\partial n} & \forall \mathbf{r} \in \partial\Omega_m \end{cases}$$

where  $\phi^-$  and  $\phi^+$  are the limiting values at the interface.

The Dirichlet boundary condition can be used:

$$\phi(\mathbf{r}) = \phi_0(\mathbf{r}) \quad \forall \mathbf{r} \in \partial\Omega$$

# Simplification and linearization of the Poisson Boltzmann equation

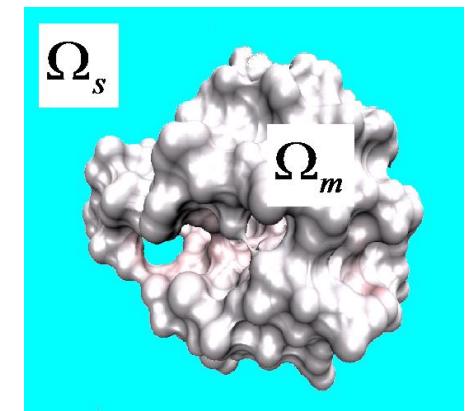
The Poisson-Boltzmann equation

$$-\nabla \cdot (\varepsilon(\mathbf{r}) \nabla \phi) - \sum_j q_j c_j e^{-\frac{q_j \phi}{kT}} = \rho_m$$

For two types of ions :  $u(r) = \frac{e\phi}{kT}$ ,

$$-\nabla \cdot (\varepsilon(\mathbf{r}) \nabla u) + \kappa^2 \sinh u = \frac{e}{kT} \rho_m$$

where  $\kappa^2 = \frac{2ec}{kT}$



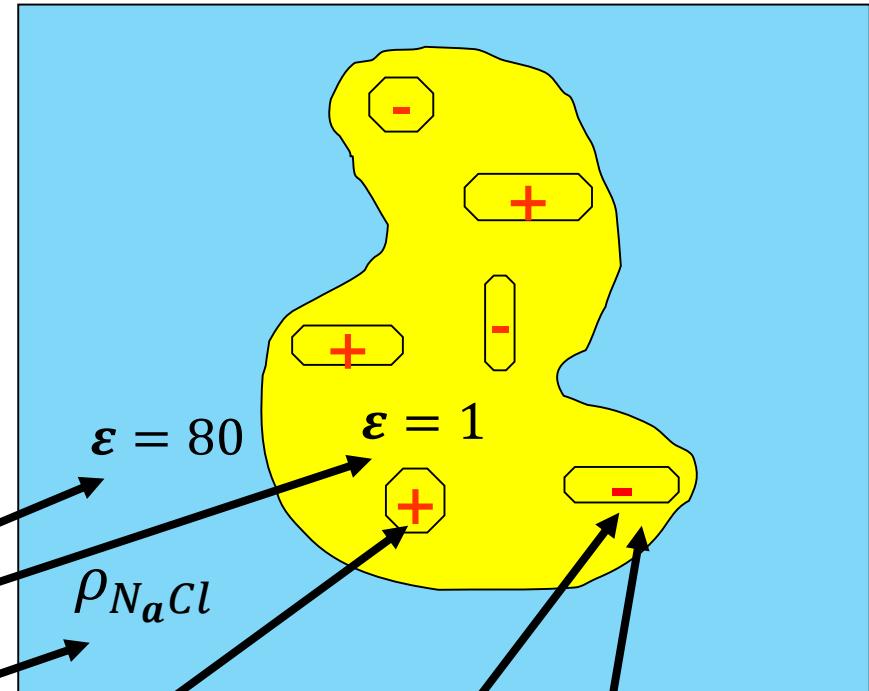
For  $u \ll 1$ , the linearized Poisson-Boltzmann equation:

$$-\nabla \cdot (\varepsilon(\mathbf{r}) \nabla u) + \kappa^2 u = \frac{e}{kT} \rho_m$$

# Multiscale: The Poisson-Boltzmann equation

- Discontinuous dielectric constant at the interface
- Non-smooth interface (geometric singularity)
- Singular charges (delta functions)

Chern et al, 2003; Geng, Yu, Wei, JCP, 2007; Geng, Zhao, JCP 2017

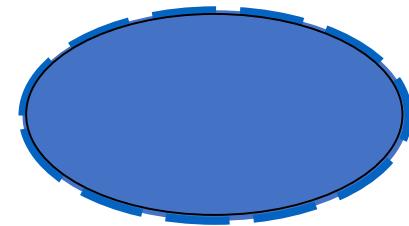
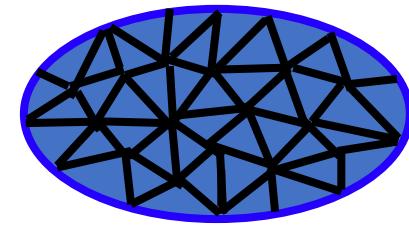
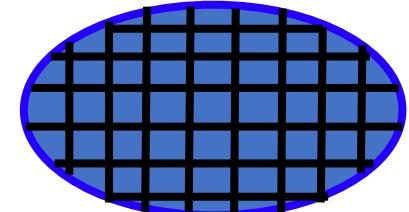


$$-\nabla \cdot (\epsilon(\mathbf{r}) \nabla \phi) = \sum_i q_i c_i e^{-\frac{q_i \phi}{kT}} + \sum_i (Q_i \delta(\mathbf{r} - \mathbf{r}_i) - \mathbf{d}_i \cdot \nabla \delta(\mathbf{r} - \mathbf{r}_i) + \Theta_i : \nabla \nabla \delta(\mathbf{r} - \mathbf{r}_i))$$

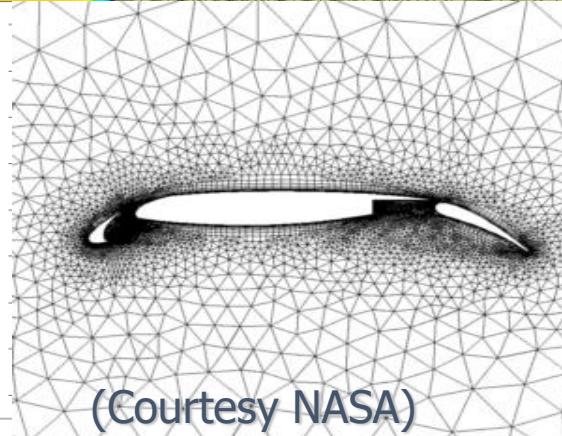
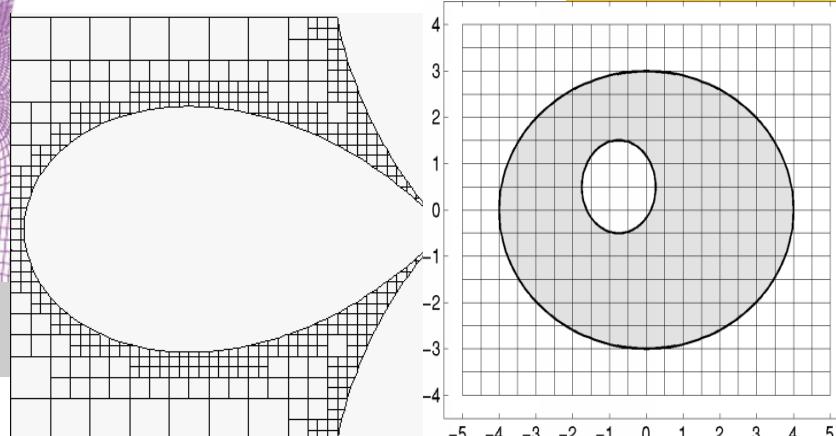
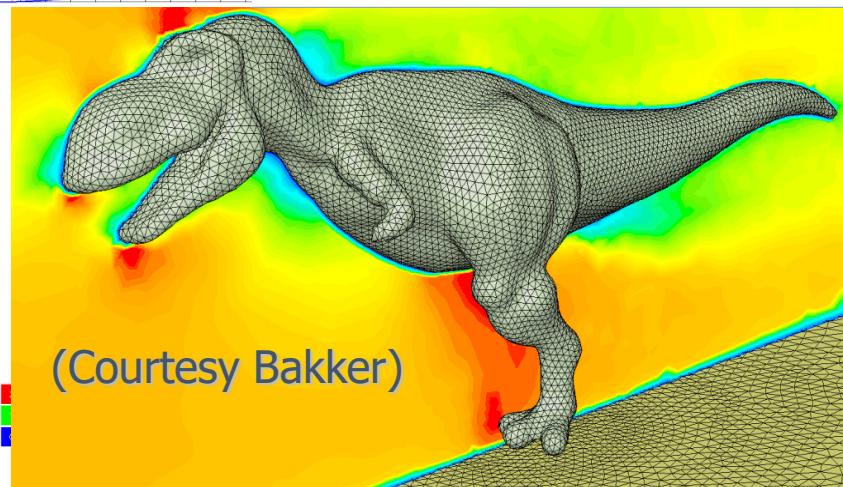
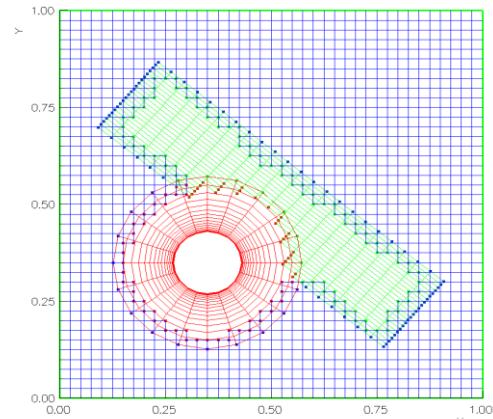
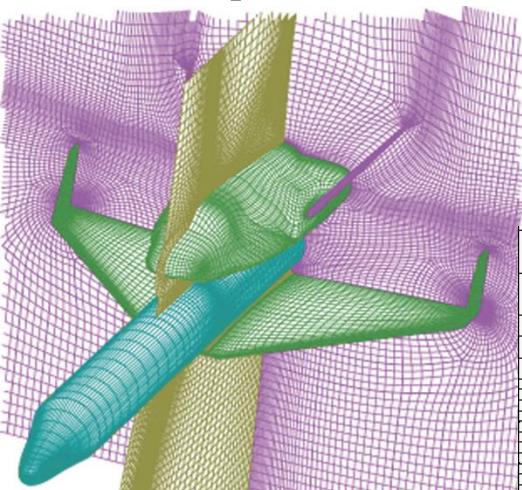
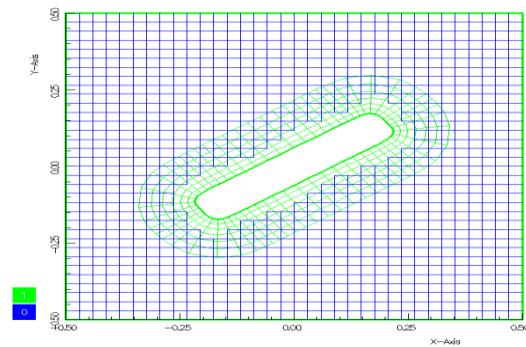
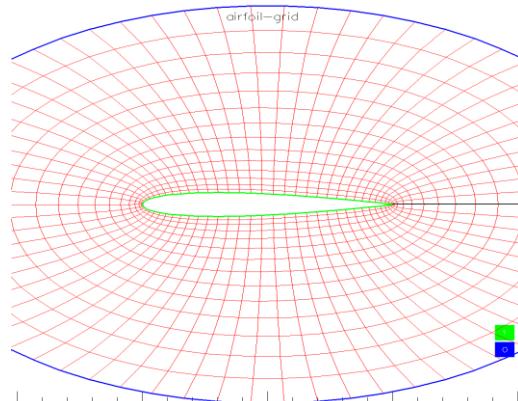
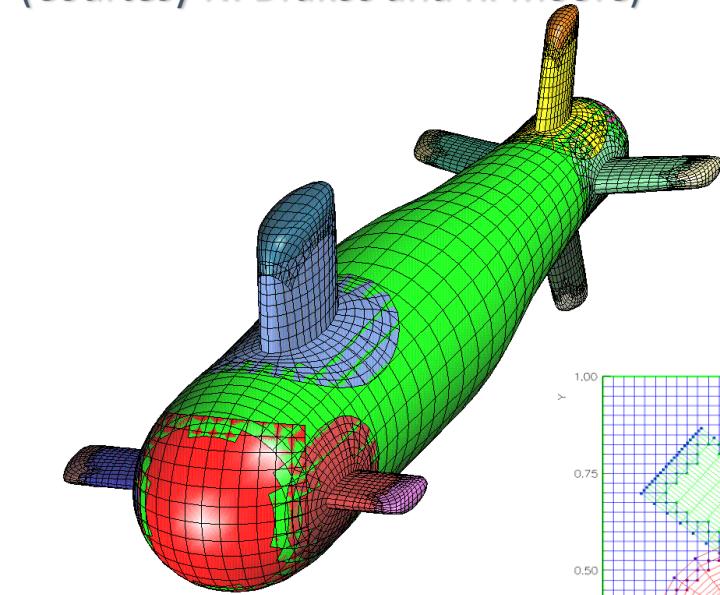
Point charge                      Charge polarization (Amoeba)

# **Major discretization methods for the Poisson-Boltzmann equation**

- Finite Difference Method: Klapper (1986); Honig (1987, 1991); Im et al (1989)...  
Software: DelPhi (Honig, Alexov, Rocchia,...); PBEQ(CHARMM, Im, Roux); FDPB(AMBER, Luo); MEAD; APBS (Baker, Holst, ...); MIBPB.
- Finite Element Method: Holst (1994); Baker et al (APBS 2001); Xie (2017),...
- Boundary Element Method/Fast multipole: Zauhar (1985); Zhou (1993); Boschitsch (2002); McCammon (2005); Lu et al (2010); Treecode: Geng & Krasny (2013);...
- Monte Carlo PBE: Hwang, Mascagni, Fenley (2001,2007,2010), ...

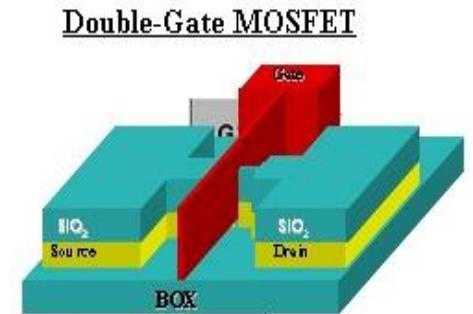
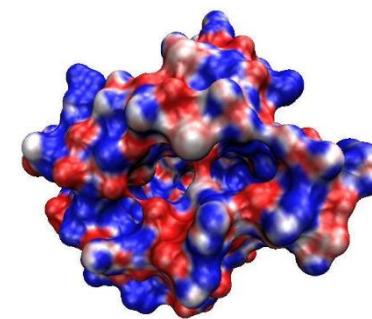
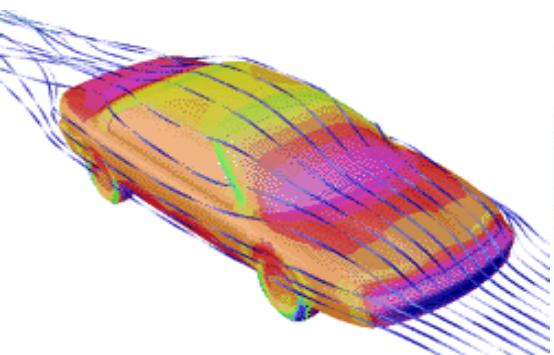
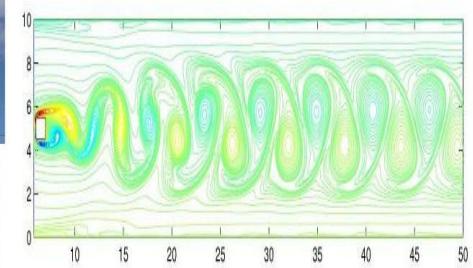
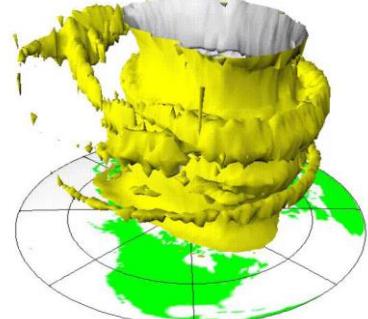
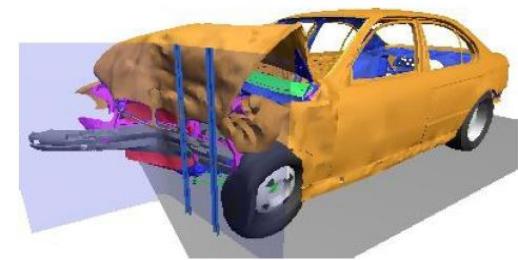
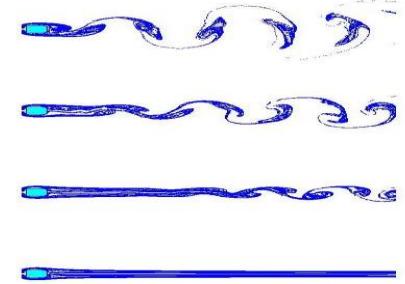
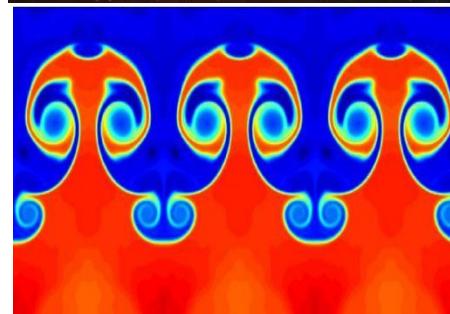
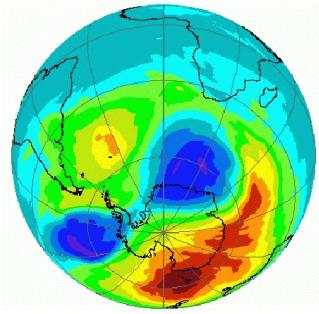
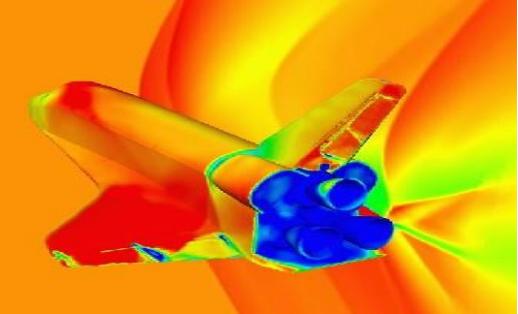
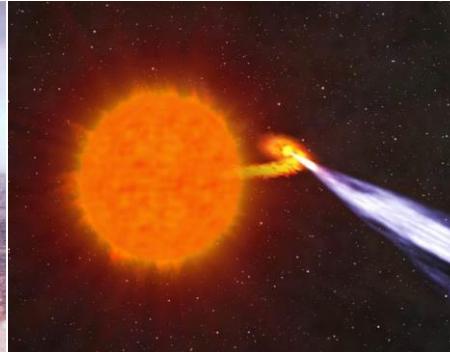
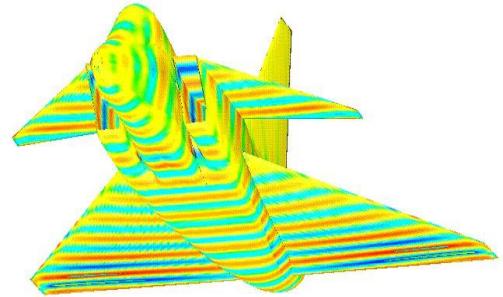


(Courtesy N. Drakos and R. Moore)



National Aeronautics  
and Space Administration

MICHIGAN STATE  
UNIVERSITY



Double-Gate MOSFET

# Method of weighted residuals

$$Lu(x) = f(x) \quad \text{or} \quad Lu(x) - f(x) = 0$$

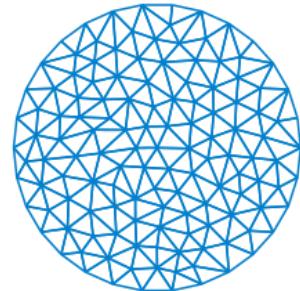
$$u(x) \approx U(x) = \sum_j^M c_j \varphi_j(x) \quad \text{for given } \varphi_j(x)$$

$$LU(x) - f(x) = R^M(x)$$

$$\int_{\Omega} R^M(x) \psi_i(x) d\Omega = 0 \quad \text{for given } \psi_i, \quad i = 1, 2, \dots, N$$

Here  $\psi_i$  and  $\varphi_i = \begin{cases} \text{polynomials} \\ \text{wavelets} \\ \text{others} \end{cases}$

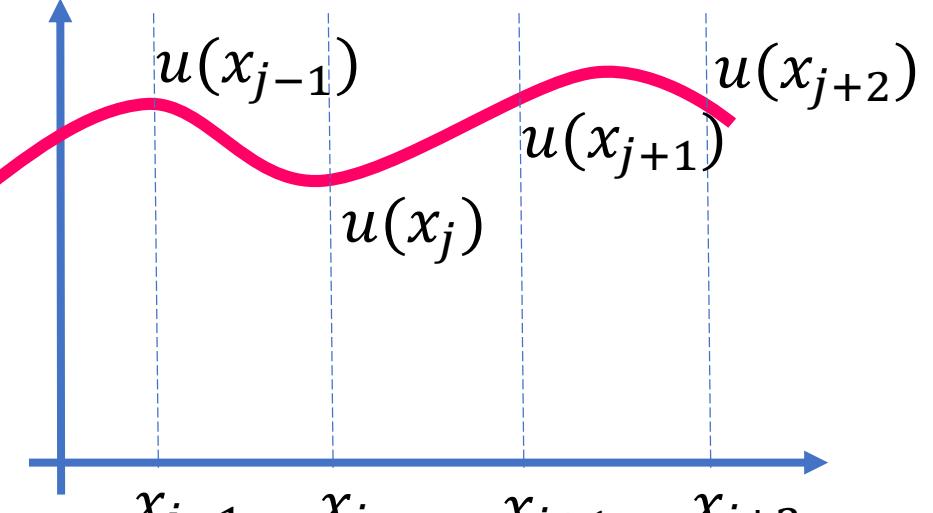
- Select  $\psi_i = \varphi_i$  and  $M = N \Rightarrow$  Galerkin finite element method
- Select  $\psi_i = \delta(x - x_i) \Rightarrow$  Collocation (finite difference) method
- The integrate equation is discretized and the resulting algebraic equations are solved to determine all  $c_j$ .



# Finite difference methods (FDMs)

$$\left. \frac{du(x)}{dx} \right|_{x=x_j} \approx \frac{u(x_{j+1}) - u(x_{j-1})}{2h}$$

$$\left. \frac{d^2u(x)}{dx^2} \right|_{x=x_j} \approx \frac{u(x_{j+1}) - 2u(x_j) + u(x_{j-1})}{h^2}$$



- Second order in accuracy
- Requires treatment at the boundary
- Requires treatment at the interface, if any.

# Main features for FDMs

## Advantages:

- Easy to achieve high order accuracy for simple geometry.
- Easy to implement boundary and interface conditions for simple geometry.
- Regular mesh leads to good time dependent behavior.
- Low condition number and memory requirements.

## Disadvantages:

- Difficult to achieve high order accuracy for complex geometry.
- Difficult to construct local adaptive refinement.
- Difficult to construct proofs due to the collocation formulation.

# Finite element methods



## Main features:

1. A problem domain is divided into elements and easy for handling complex geometry and interface.
2. Weak integral formulation is used and easy to prove the convergence.
3. Governing equations are transformed into approximate algebraic equations for each element in the mesh and assembled based on the element connectivity.

# Main features for FEMs

## Advantages:

Adaptive to complex geometry (h-p refinements).

- Flexible for boundary and interface conditions.
- Good for time independent problems.
- Easy to construct proofs due to the weak formulation.

## Disadvantages:

- Time-stepping is small for time dependent problems.
- High condition number.
- Large memory storage requirement.
- Difficult to achieve high-order accuracy for complex geometry or low solution regularity.

# Boundary element methods

$$-\nabla \cdot (\varepsilon \nabla u) + \kappa^2 u = \frac{e}{kT} \sum_j Q_j \delta(\mathbf{r} - \mathbf{r}_j)$$

$$u_m(x) = \oint_{\Gamma} \left[ G_m(x, y) \frac{\partial u_m(y)}{\partial y} - \frac{\partial G_m(y)}{\partial y} u_m(y) \right] d\Gamma_y + \frac{e}{kT} \sum_j Q_j G_m(x, y_j)$$

$$u_s(x) = \oint_{\Gamma} \left[ G_s(x, y) \frac{\partial u_s(y)}{\partial y} - \frac{\partial G_s(y)}{\partial y} u_s(y) \right] d\Gamma_y$$

where  $G_m(x, y) = \frac{1}{4\pi|x-y|}$  and  $G_s(x, y) = \frac{e^{-\kappa|x-y|}}{4\pi|x-y|}$  are fundamental solutions.

# Main features for BEMs

## Advantages:

- The dimension is reduced due to the integral equation formulation --- low storage and low computational cost.
- Adaptive for boundary and interface conditions.
- Exact for far field boundary conditions and analytical for charge source.
- Relatively easy to construct proofs due to the weak formulation.

## Disadvantages:

- Requiring the fundamental solution.
- Small time-stepping due to irregular mesh.
- Difficult for nonlinear problems.
- Difficult to achieve high order accuracy for complex geometry.

# **Major iterative methods for the Poisson-Boltzmann equation**

- Jacobi method
- Successive over-relaxation (SOR)
- Bi-Conjugate gradient (BiCG)
- Incomplete LU factorization
- Generalized minimal residual method (GMRES)

Solver library using preconditioner:

- PETSc (<http://www.mcs.anl.gov/petsc/petsc-as/>)
- SLATEC  
([http://people.sc.fsu.edu/~burkardt/f\\_src/slatec/slatec.html](http://people.sc.fsu.edu/~burkardt/f_src/slatec/slatec.html))

# Elliptic equation with material interface

Elliptic equation

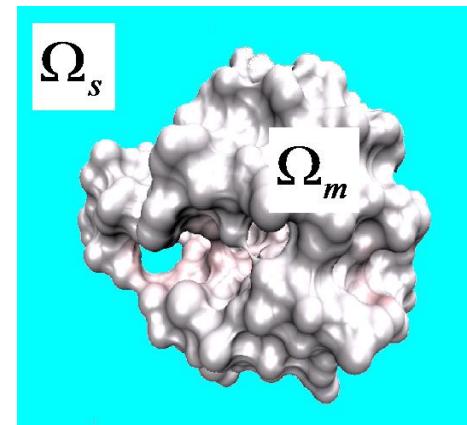
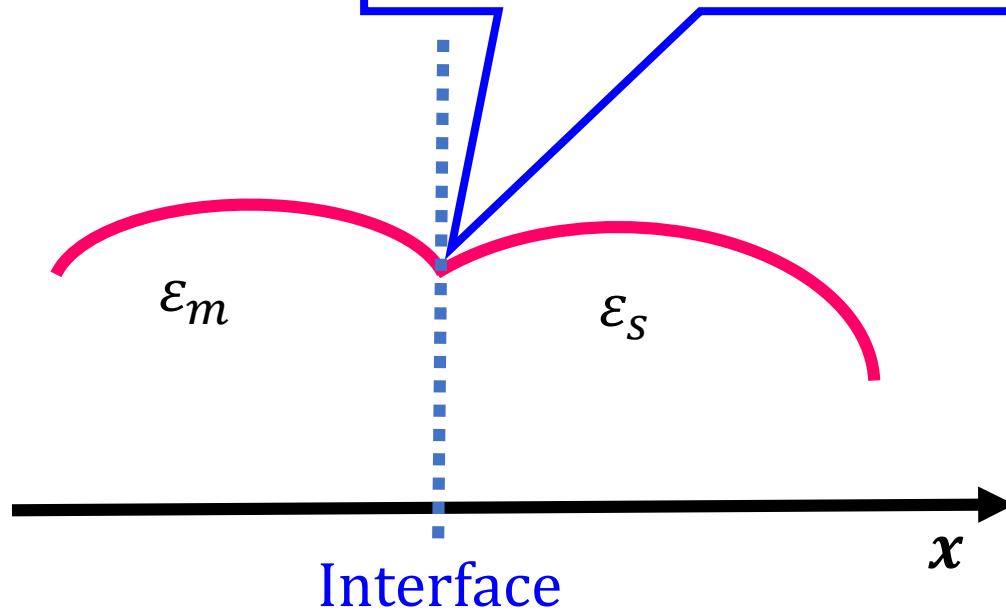
$$-\nabla \cdot (\varepsilon(\mathbf{r}) \nabla u) = f$$

with continuity condition at the interface:

$$u^- = u^+$$

$$\varepsilon_m \frac{du^-}{dx} = \varepsilon_s \frac{du^+}{dx}$$

Solution loses its regularity  
across the interface



# **Elliptic PDE with discontinuous coefficients in mathematical community of finite difference**

- Immersed boundary method (IBM, Peskin, 1977; Lai 2000s)
- Immersed interface method (IIM, LeVeque and Li, 1994; Li and Ito, 2001, Deng, Ito and Li, 2003; T Lin; many others)
- Ghost fluid method (GFM, Fedkiw and Osher, 2000)
- Integral equation approach (Mayo, 1984, 1992)
- Delta function (Tornberg and Engquist, 2004)
- Matched interface and boundary (MIB, My group, 2002-present;  
Independent efforts: S Zhao; YC Zhou; WH Geng)

# Matched interface & boundary (MIB)

Local finite difference scheme:

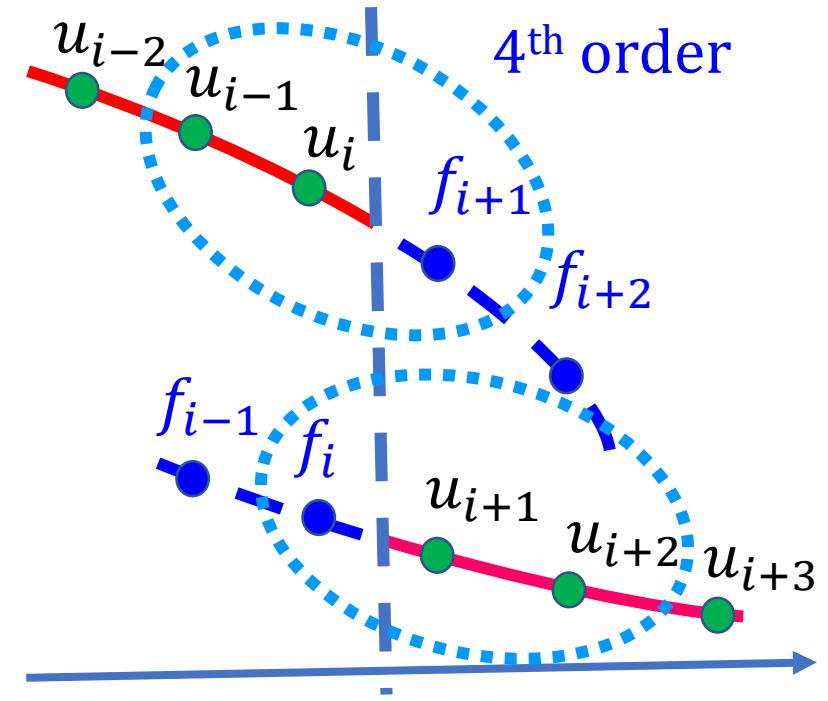
$$\frac{d^2u}{dx^2} \bigg|_{x=x_i} \approx \frac{u_{i-1} - 2u_i + f_{i+1}}{h^2}$$

$$\frac{d^2u}{dx^2} \bigg|_{x=x_{i+1}} \approx \frac{f_i - 2u_{i+1} + u_{i+2}}{h^2}$$

Fictitious values  $f_{i+1}$  and  $f_i$  are determined by two interface conditions:

$$u^- - u^+ = \xi$$

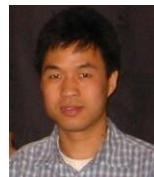
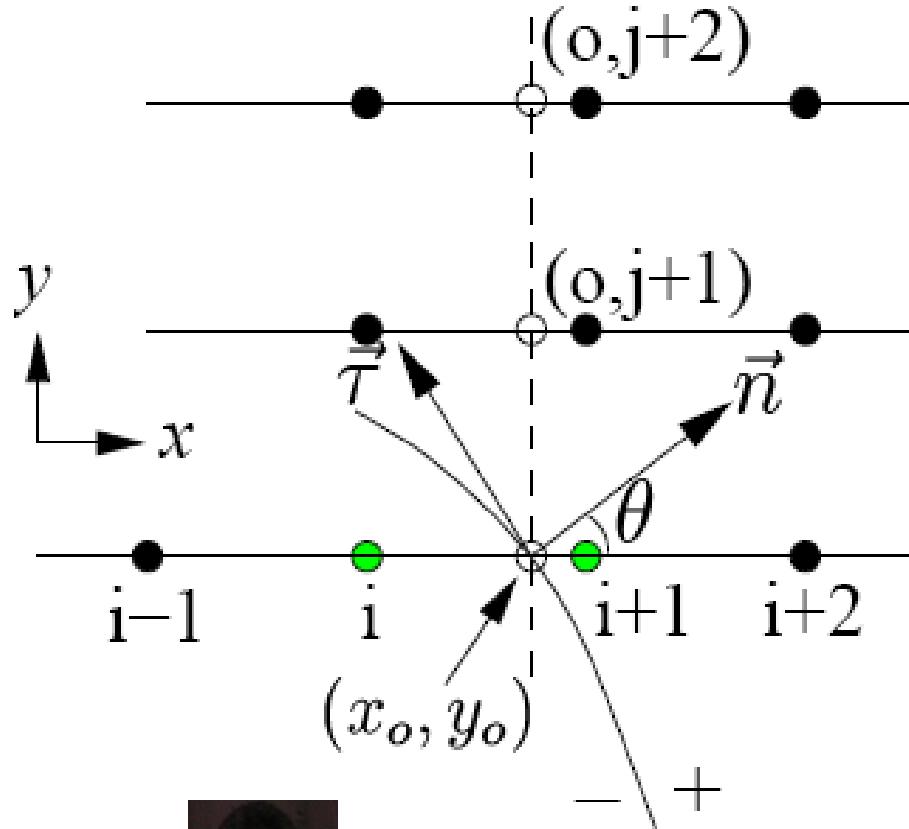
$$\varepsilon_m \frac{du^-}{dx} - \varepsilon_s \frac{du^+}{dx} = \zeta$$



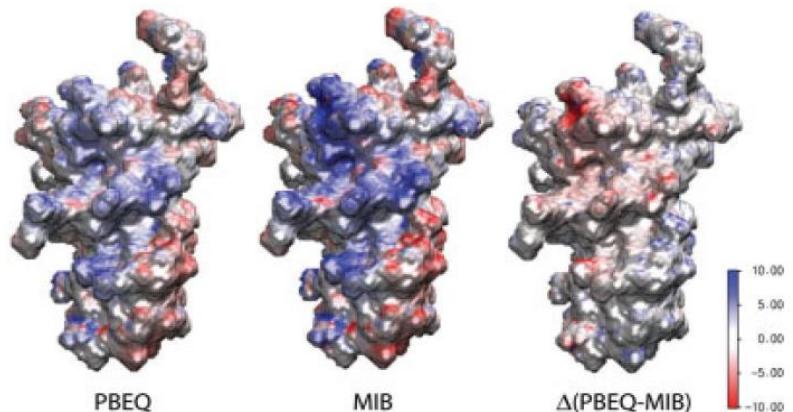
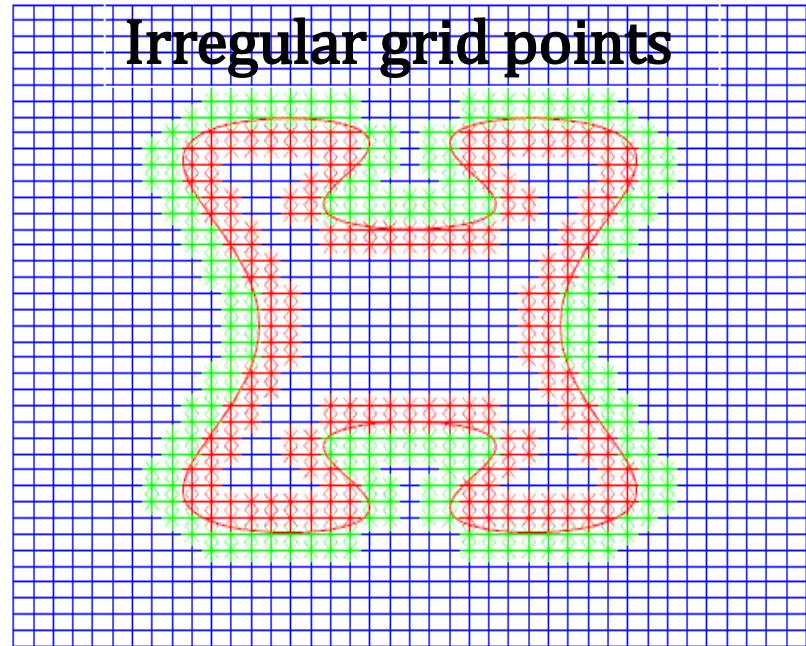
(Zhao and Wei, JCP, 2004)

# 2D and 3D 2<sup>nd</sup> order Curved interface scheme

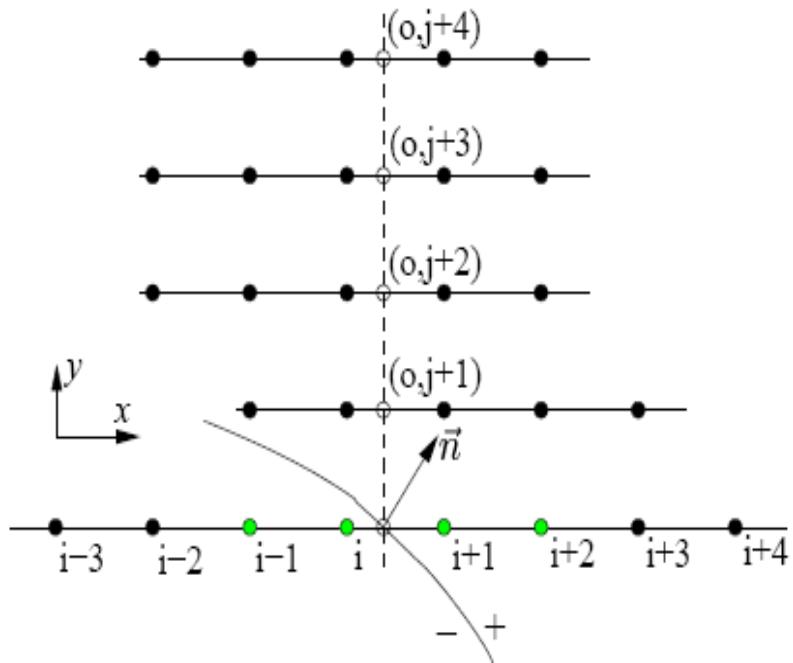
One local coordinate is set at each intersecting point of the mesh and the interface.



(Zhou, Zhao, Feig and Wei, JCP 2006;  
Zhou, Feig and Wei, 2008)

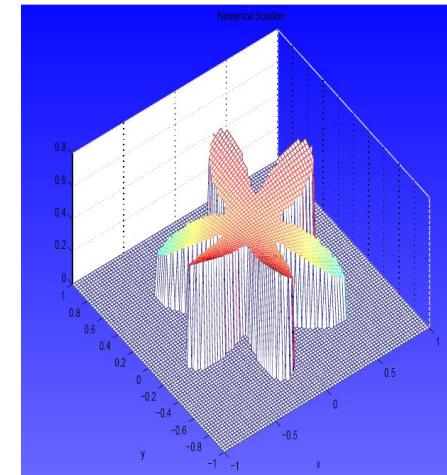
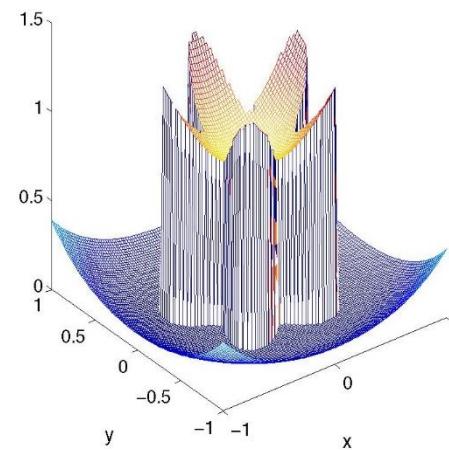


# Numerical Validation (4<sup>th</sup> order MIB scheme)

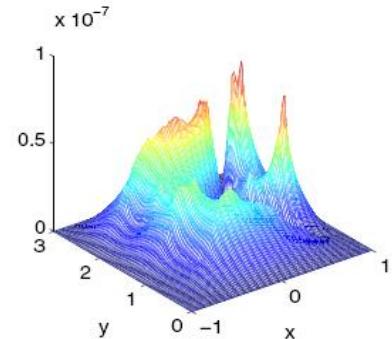
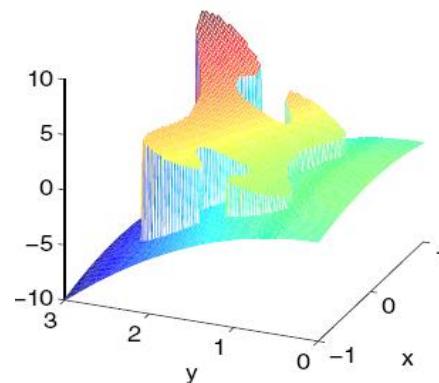
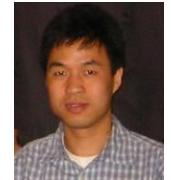


$$u(x, y) = \begin{cases} e^x(y^2 + x^2 \sin(y)), & (x, y) \in \Omega^- \\ -(x^2 + y^2), & (x, y) \in \Omega^+ \end{cases}$$

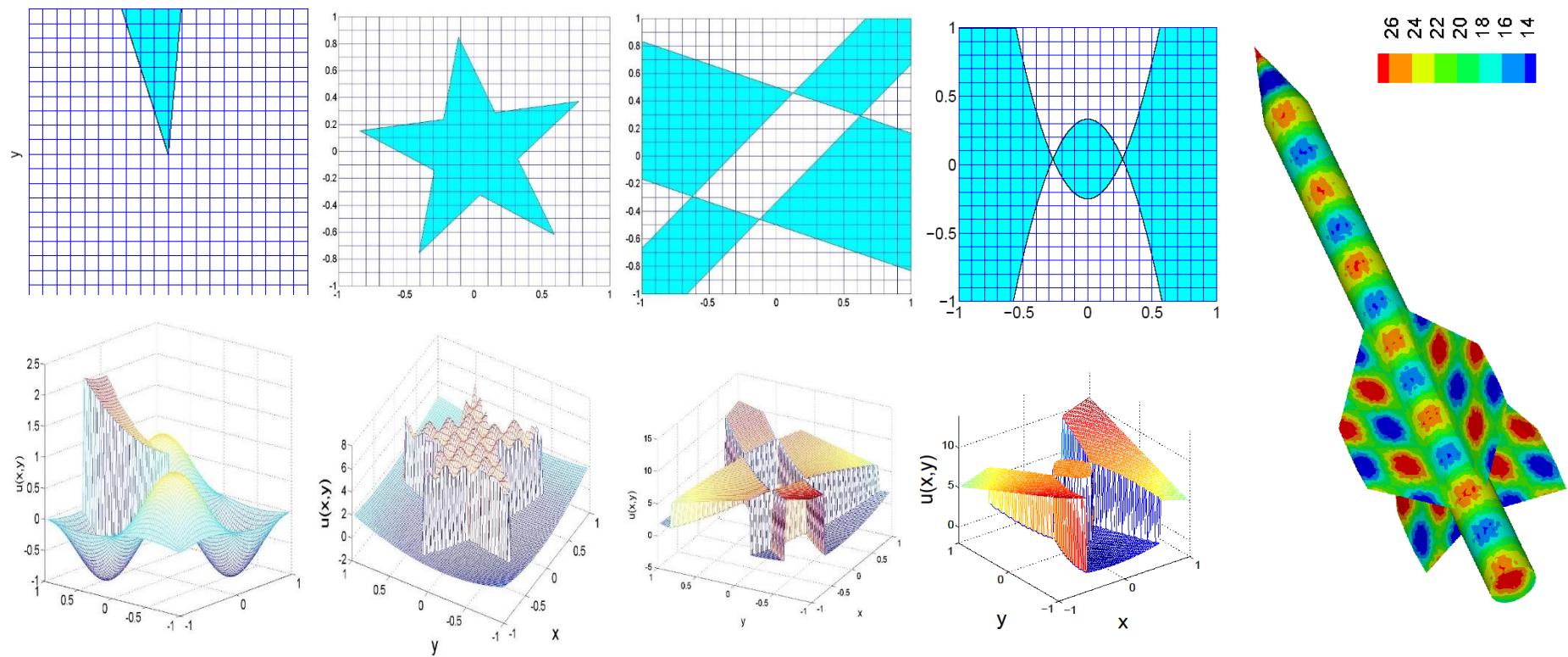
$$\varepsilon = \begin{cases} 1, & r \leq 0.5 \\ 10, & r > 0.5 \end{cases}$$



(Zhou, Zhao, Feig and Wei, JCP 2006)



# Elliptic equations with sharp-edged interfaces (2<sup>nd</sup> order)

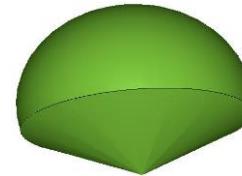
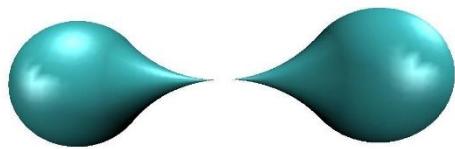


(Yu, Zhou and Wei, JCP, 2007; Yu and Wei, JCP, 2007)

# MIB accuracies

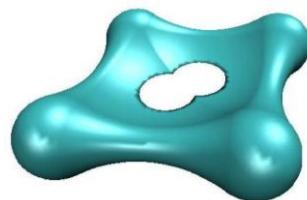


Sining Yu



Mesh size ( $\text{\AA}$ )	Error	Order
0.5	3.91E-2	-
0.25	3.66E-3	3.42
0.125	9.31E-4	1.98

Mesh	Error	Order
0.2	7.09E-2	-
0.1	3.67E-3	4.27
0.05	2.00E-4	4.20



(Yu and Wei, JCP, 2007)

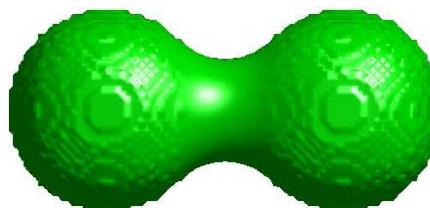


Mesh size ( $\text{\AA}$ )	Error	Order
0.5	1.30E-2	-
0.25	3.80E-3	1.77
0.125	9.17E-4	2.05

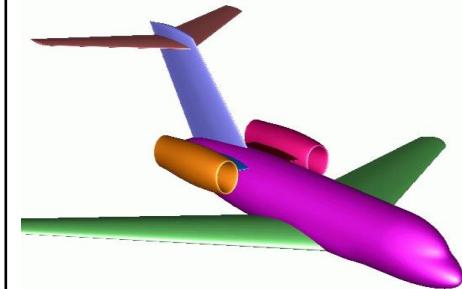
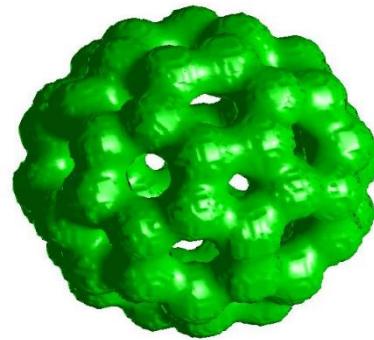
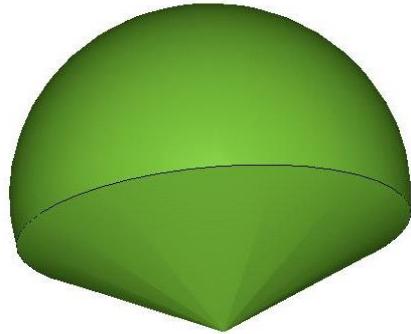
Mesh size	Error	Order
0.2	5.3E-7	-
0.1	5.4E-9	6.6
0.05	5.3E-11	6.7

# MIB capability

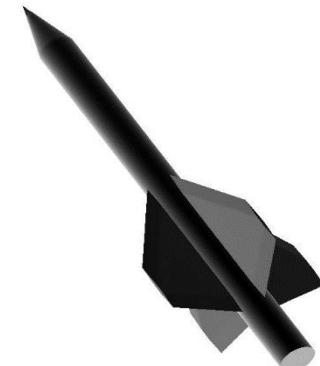
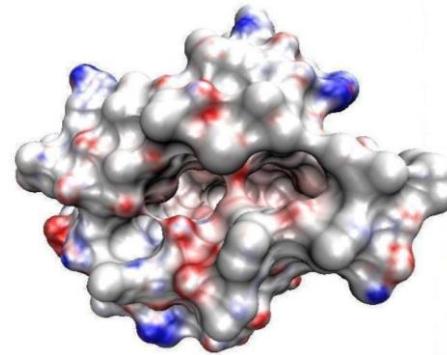
6th order accuracy



4th order accuracy



2nd order accuracy



# MIB-Green's function for singular charges

(Chern, Liu, and Wang, MAA, 2003; Chen, Holst and Xu, SIAM JNA, 2007;  
Geng and Zhao, 2017; Xie, 2014, ...)

Decompose the solution into singular and regular charges

$$\phi = \bar{\phi} + \tilde{\phi}$$

where  $\bar{\phi}(\mathbf{r}) = \begin{cases} \phi^*(\mathbf{r}) + \phi^0(\mathbf{r}), & \mathbf{r} \in \Omega_m \\ 0, & \mathbf{r} \in \Omega_s \end{cases}$

$\phi^*(\mathbf{r})$  is the potential in the free space induced by the singular charges

$$\phi^*(\mathbf{r}) = \sum_i^M \frac{z_i}{4\pi\varepsilon_m |\mathbf{r} - \mathbf{r}_i|}$$

$\phi^0(\mathbf{r})$  is the harmonic function in  $\Omega_m$  satisfying

$$\begin{cases} \Delta\phi^0(\mathbf{r}) = 0, & \mathbf{r} \in \Omega_m \\ \phi^0(\mathbf{r}) = -\phi^*(\mathbf{r}), & \mathbf{r} \in \partial\Omega_m \end{cases}$$



(Geng, Yu and Wei, JCP, 2007)

## MIB-Green's function for singular charges

$\bar{\phi}(\mathbf{r})$  is the solution to the PBE satisfying the following jump conditions:

$$[\bar{\phi}]_{\Gamma} = 0 \text{ and } [\varepsilon \bar{\phi}_n]_{\Gamma} = -\varepsilon \nabla(\phi^*(\mathbf{r}) + \phi^0(\mathbf{r})) \cdot n \Big|_{\Gamma}$$

The equation for the corrected potential  $\tilde{\phi}(\mathbf{r})$  is

$$-\nabla \cdot (\varepsilon(\mathbf{r}) \nabla \tilde{\phi}) + \bar{\kappa}^2 \sinh(\bar{\phi} + \tilde{\phi}) = 0$$

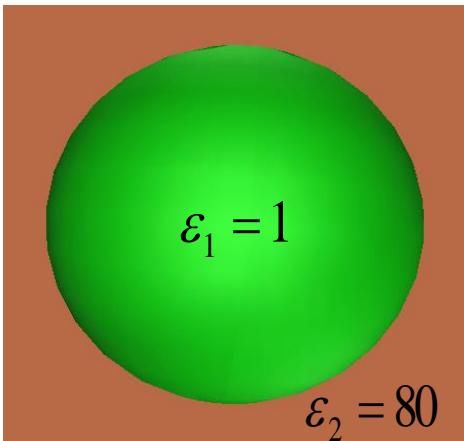
with jump conditions

$$[\tilde{\phi}]_{\Gamma} = 0 \text{ and } [\varepsilon \tilde{\phi}_n]_{\Gamma} = -[\varepsilon \bar{\phi}_n]_{\Gamma} = \varepsilon \nabla(\phi^*(\mathbf{r}) + \phi^0(\mathbf{r})) \cdot n \Big|_{\Gamma}$$

# Kirkwood sphere with a unit charge

(Geng, Yu and Wei, JCP, 2007)

$$\Delta G_{\text{solvation}} = \Delta G_{\text{solution}} - \Delta G_{\text{vacuum}}$$



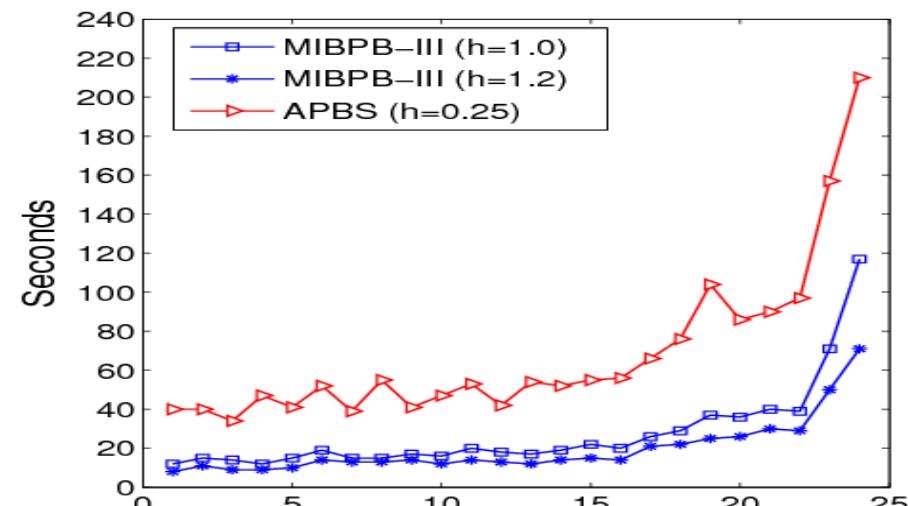
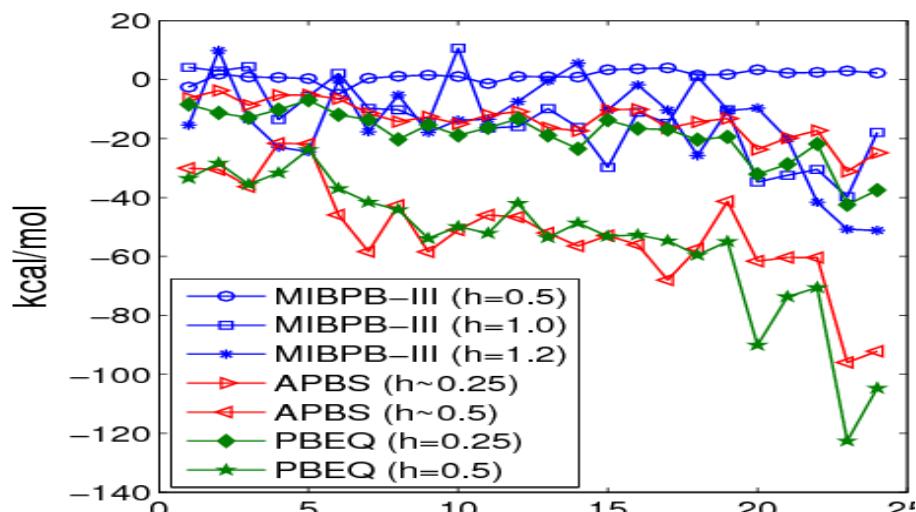
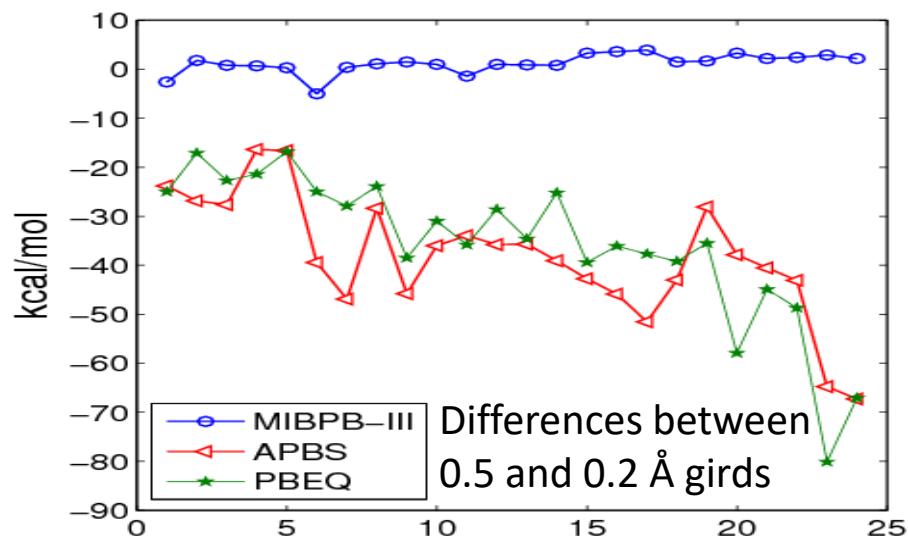
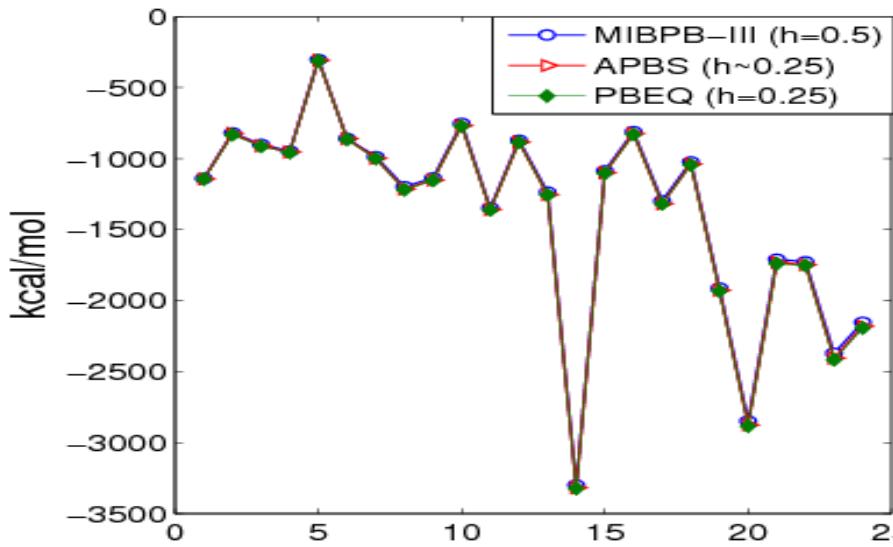
Solvation  
energy  
(Surface  
potential  
error)

Mesh (Å)	PBEQ	APBS	MIB	MIB-GF
1.0	-83.57 (25.31)	-83.68 (25.23)	-83.68 (7.31)	-81.95 (0.16)
0.5	-85.78 (17.05)	-85.85 (17.06)	-81.97 (2.56)	-81.98 (0.02)
0.2	-82.84 (7.51)	-82.58 (7.50)	-81.98 (0.30)	-81.98 (0.005)
0.1	-82.49 (3.84)	-82.27 (3.83)	-81.98 (0.04)	-81.98 (0.001)
0.05	-82.20 (1.94)	-82.03 (1.89)	-81.98 (0.01)	-81.98 (0.0003)

Exact solution: -81.980 kcal/mol

BEM: -82.12 (1280 elements)

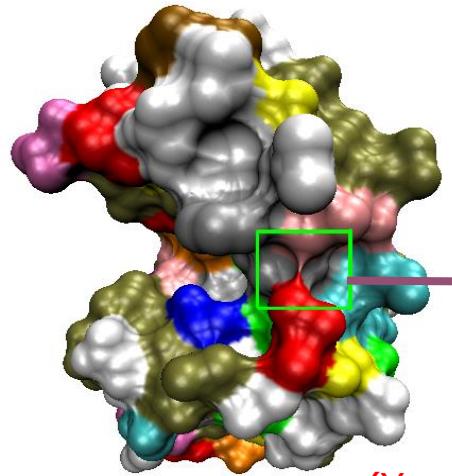
# Comparison of solvation free energies



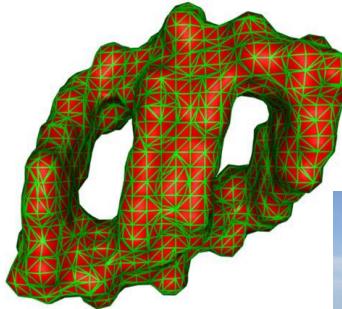
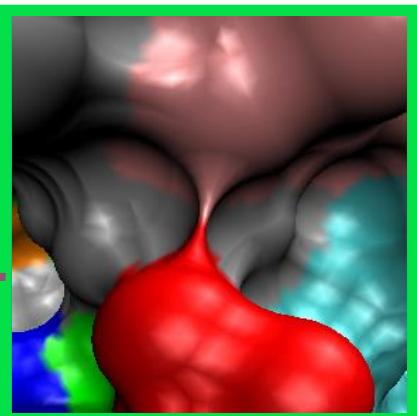
From left to right: 1ajj, 1vii, 2erl, 1bbl, 1cbn, 2pde, 1sh1, 1fca, 1ptq, 1bor, 1uxc, 1vjw, 1fxd, 1hpt, 1mbg, 1bpi, 1r69, 451c, 1neq, 1a2s, 1svr, 1frd, 1a63, and 1a7m

(Geng, Yu and Wei, JCP, 2007)

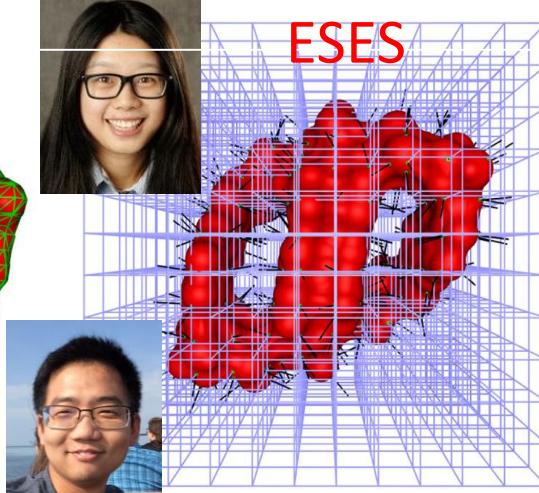
# MIBPB for solving the Poisson equation with protein interface



(Yu, Geng, Wei, JCP 2007)

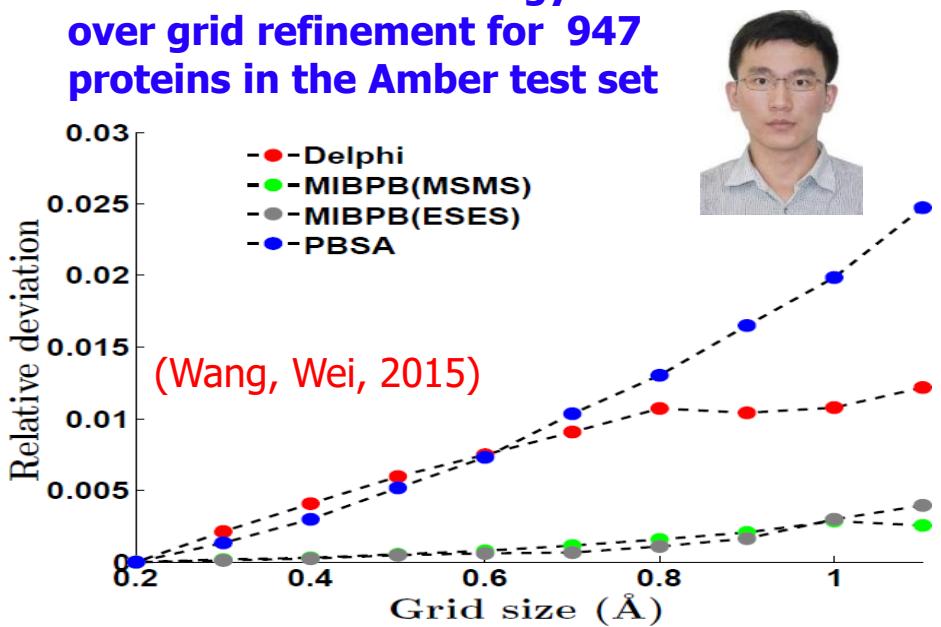


ESES



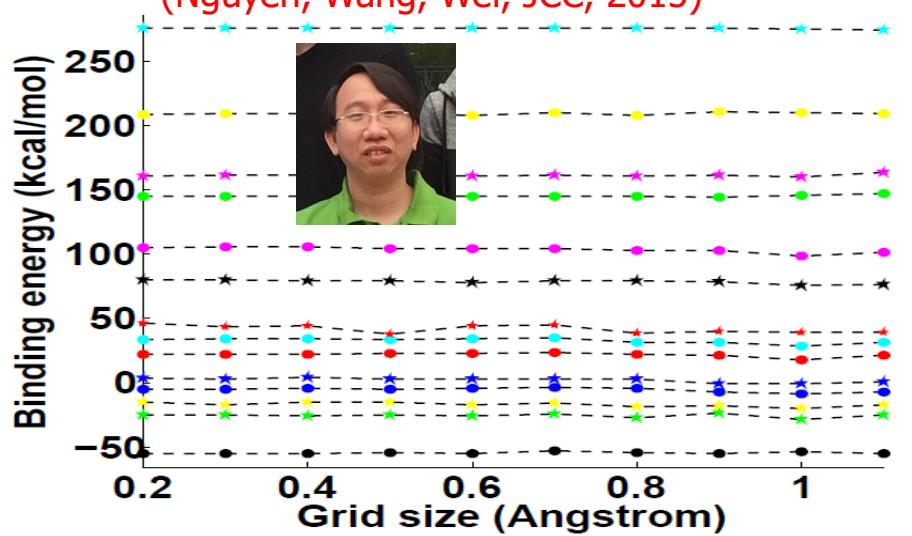
(Liu, Wang, Zhao, Tong, Wei, JCC 2017)

## Relative solvation energy deviations over grid refinement for 947 proteins in the Amber test set



## Electrostatic binding energies of 14 RNA-protein complexes over grid refinement

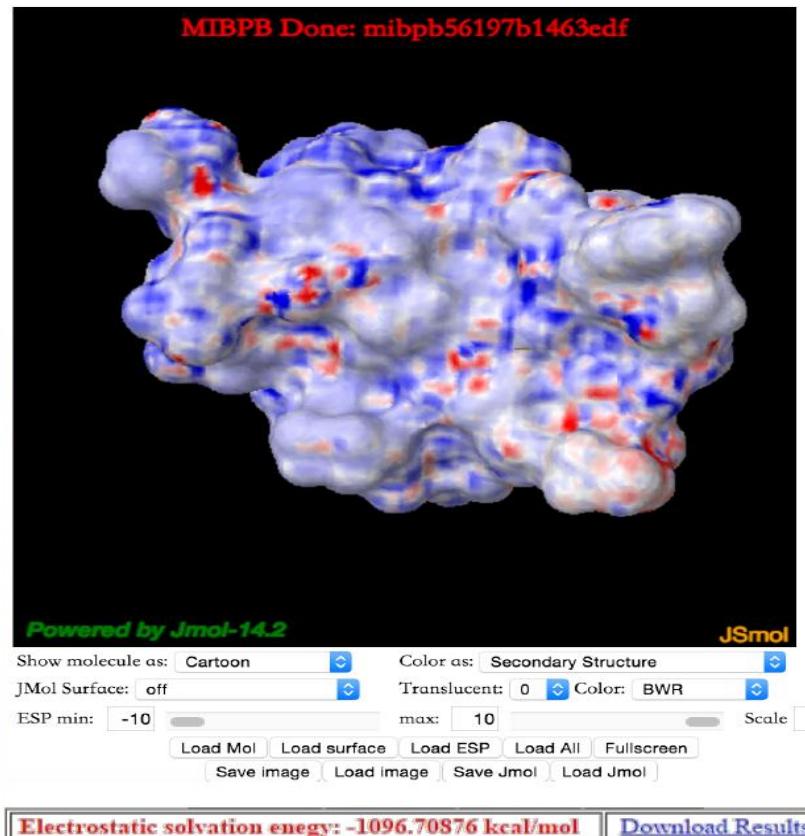
(Nguyen, Wang, Wei, JCC, 2015)





MIBPB is a software package for obtaining electrostatic potential and solvation free energy via solving the Poisson-Boltzmann (PB) equation. It makes use of the second order convergent MIB technique and is essentially grid independent. Its mean relative error is less than 0.5% for about 1000 test proteins when the grid size is refined from 1.1 to 0.2 Angstrom. [\[User Manual\]](#)

Input_File:		<input checked="" type="radio"/> PDB ID: <input type="text"/> Chains: * <input type="radio"/> User File: <input type="button" value="选择文件"/> 未选择任何文件 *
MIBPB Options		1.0 Interior Dielectric
		80 Outerior Dielectric
		0.8 Grid Resolution
		0.0 Ion Strength
		1.4 Surface Probe Radius for MIBPB
		Linearized PB: <input checked="" type="checkbox"/> Yes
		Simplified Solver: <input type="checkbox"/> Yes
Resulting Surface Options <input checked="" type="checkbox"/> Yes		1.4 Probe Radius
		0.8 Grid Resolution
		2.0 Grid Extension
PDB2PQR Options <input checked="" type="radio"/> Yes		Force Field: AMBER
		Protonation: at pH: 7.0 by: PROPKA
		Remove Water: <input checked="" type="checkbox"/> Yes
		Remove Hydrogen: <input type="checkbox"/> Yes
		Only assign charges and radius: <input type="checkbox"/> Yes
Small molecule to PQR Options <input checked="" type="radio"/> Yes		Charge Type: AM1-BCC
		Radius Type: mbondi
pKa Calculation <input type="checkbox"/> Yes		Residue Type: ASP
		Residue ID: <input type="button" value="Analyze online PDB"/>
Job Title: <input type="text"/>		
User Email: <input type="text"/>		
<input type="button" value="Default"/> <input type="button" value="Submit"/> <input type="button" value="Clear Job"/>		



#### DOWNLOAD

- For academic/governmental users, you may download and use MIBPB for free under a license agreement. Please follow the instructions below to [register with us and download MIBPB](#).
- For industrial/commercial users, a moderate license fee may apply. Please contact us directly at [wei@math.msu.edu](mailto:wei@math.msu.edu).
- If you have any questions or bugs to report, please feel free to contact: [wangbaonj@gmail.com](mailto:wangbaonj@gmail.com)

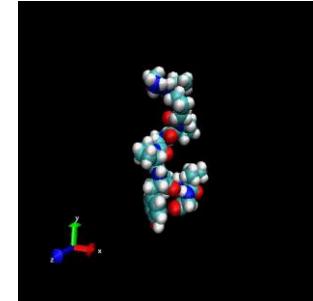
(Wang, Zhao, Wei, 2015)



# **Main features of the marched interface & boundary (MIB) method**

- High-order subgrid finite difference method for arbitrarily complex interfaces
- Simplify the 3D problem into 1D-like ones
- Disassociation between the discretization and the enforcement of jump condition
- Disassociation between the discretization and the domain extension
- Repeated use of only lowest order jump conditions

# MIBPB based multiscale molecular mechanics



Newton's second law for the motion of the  $i$ th atom :

$$m_i \frac{d^2}{dt^2} \mathbf{r}_i = \mathbf{F} = -\nabla U(\mathbf{r}_1(t), \mathbf{r}_2(t), \dots, \mathbf{r}_n(t))$$

Potential:

$$\begin{aligned} U &= U_{\text{bonded}} + U_{\text{non-bonded}} \\ U_{\text{bonded}} &= U_{\text{bonds}} + U_{\text{angle}} + U_{\text{dihedral}} \end{aligned}$$

The goal is to reduce the number of degrees of freedom by treating the solvent (water) molecules as a dielectric continuum. The PB model is used to take care the water effect.

## **PB based multiscale molecular dynamics**

(All smooth surface approaches)

Sharp and Honig (1990) --- (free energy functional)

Gilson, Davis, Luty and Mccammon (1993) (forces)

Im, Beglov and Roux (1998) (CHARMM-PB-MD)

Lu, Chen, Wang and Xu (2002)

Lu and Luo (2003) (AMBER-PB-MD)

Luo, David and Gilson (2004)

Prabhu, Zhu and Sharp (2004).....

**Problems with smooth surface MD:**

Unstable with sharp dielectrics

Overestimating solvation free energies and forces

Underestimating hydrophobicity → unfolded  
structures

# What are involved in PB based molecular dynamics?

Electrostatic forces due to the exclusion of water:

- Reaction field force,
- Dielectric boundary force, and
- Ionic boundary force.
- Other (original) forces:  
Bonding: Bond, angle and dihedral angle (AMBER)  
Non-bonding: Electrostatics and van der Waals (AMBER)

# PB based forces

Reaction field force:

$$\mathbf{F}^{\text{RF}} = 4\pi \sum_{j=1}^N Q_j \left[ (\nabla \phi_m)_{\mathbf{r}=\mathbf{r}_j} - (\nabla \phi_{\text{vacuum}})_{\mathbf{r}=\mathbf{r}_j} \right]$$

Dielectric boundary force:

$$\mathbf{F}^{\text{DB}} = \frac{1}{2} \int_{\Gamma} d\mathbf{S} [\varepsilon_s |\nabla \phi^+|^2 - \varepsilon_m |\nabla \phi^-|^2]$$

Ionic boundary force:

$$\mathbf{F}^{\text{IB}} = kT \int_{\Gamma} d\mathbf{S} \sum_i c_i \left( e^{-\frac{q_i \phi^+}{kT}} - 1 \right)$$



(Geng & Wei, JCP, 2011)

## Difficulties due to the sharp interface

Dielectric boundary force (and Ionic Boundary force):

$$\mathbf{F}^{\text{DB}} = \frac{1}{2} \int_{\Gamma} d\mathbf{S} [\varepsilon_s |\nabla \phi^+|^2 - \varepsilon_m |\nabla \phi^-|^2]$$

- (1) Computing the gradients at the interfaces (MIBPB)
- (2) Surface integration in Cartesian grid
- (3) Redistribute the force on the MS to involved atoms

(Geng & Wei, JCP, 2011)

# Forces of two separating atoms

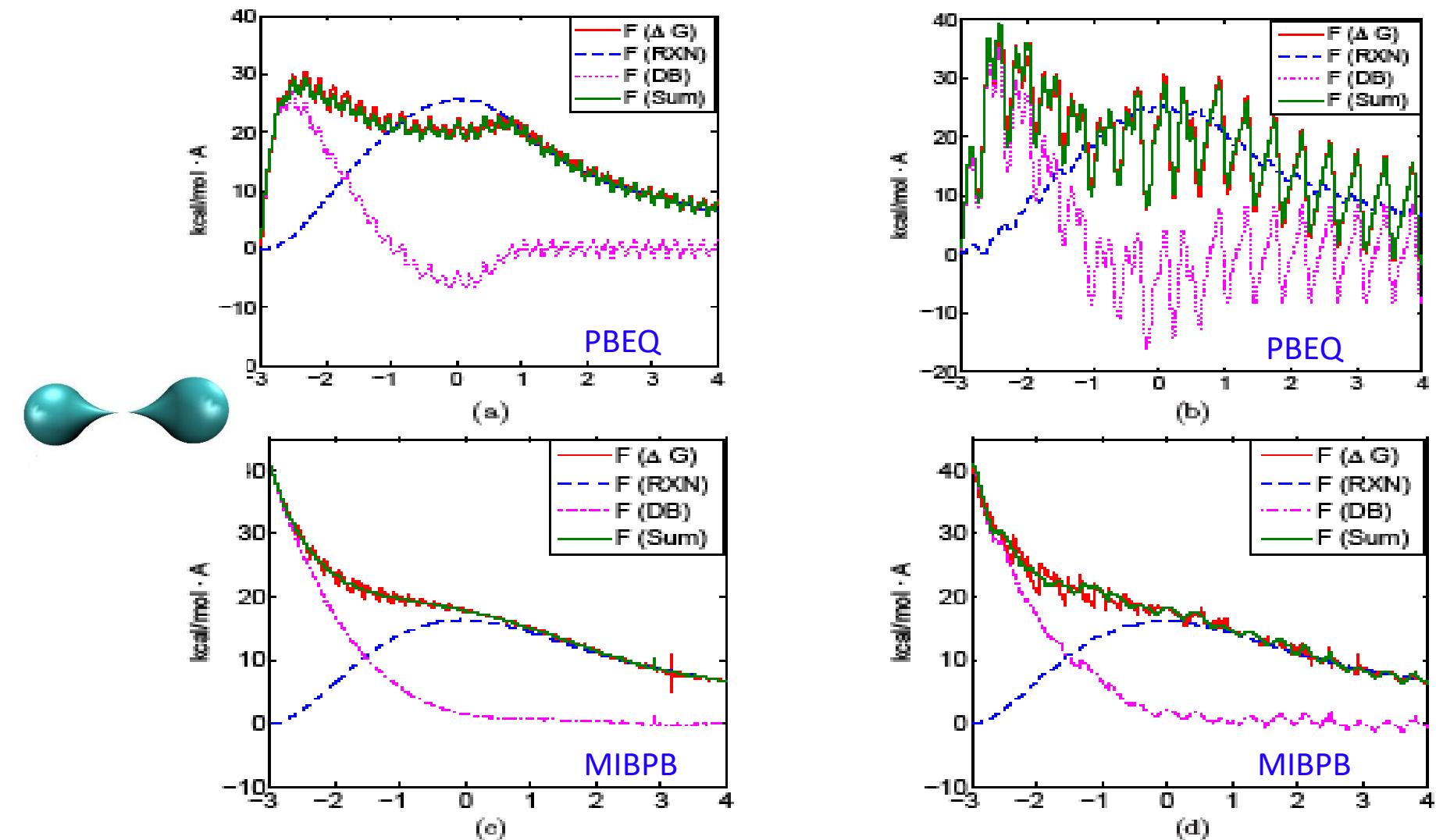
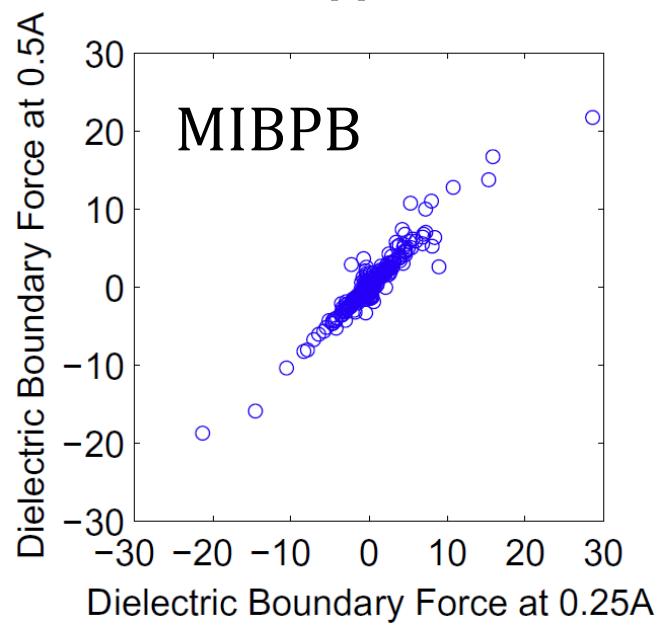
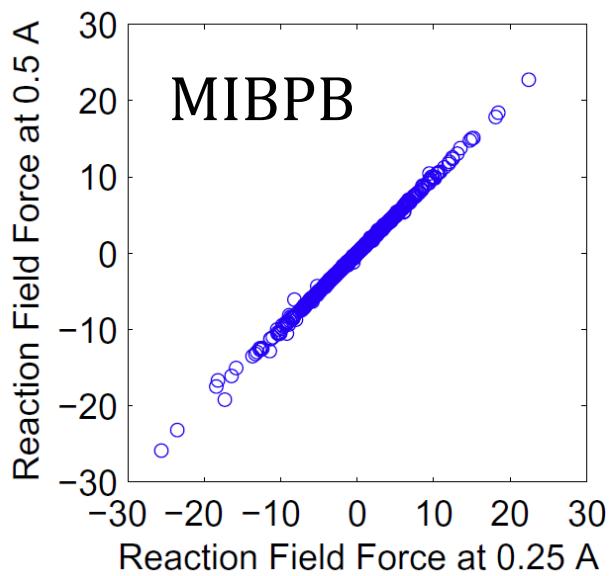
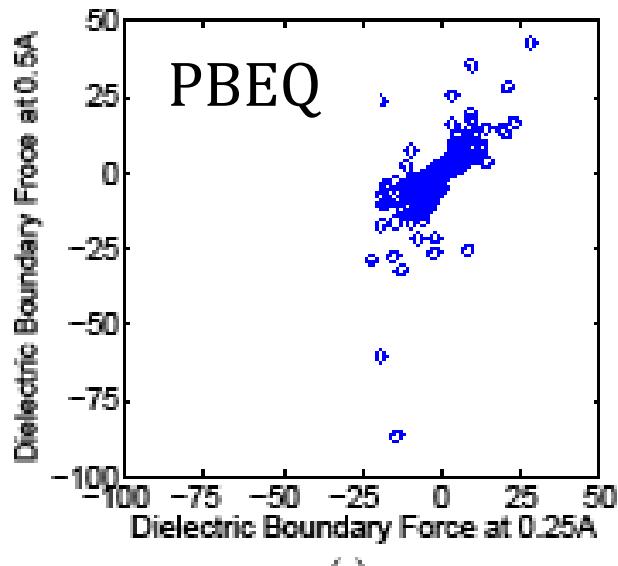
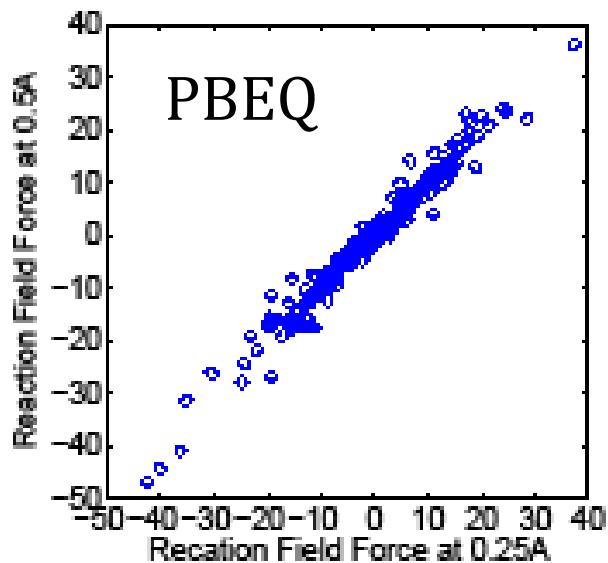
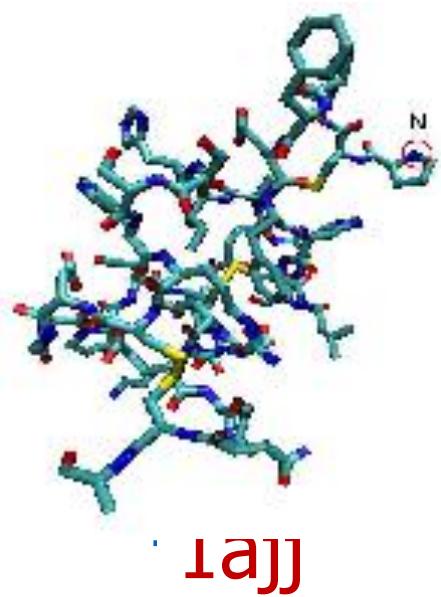
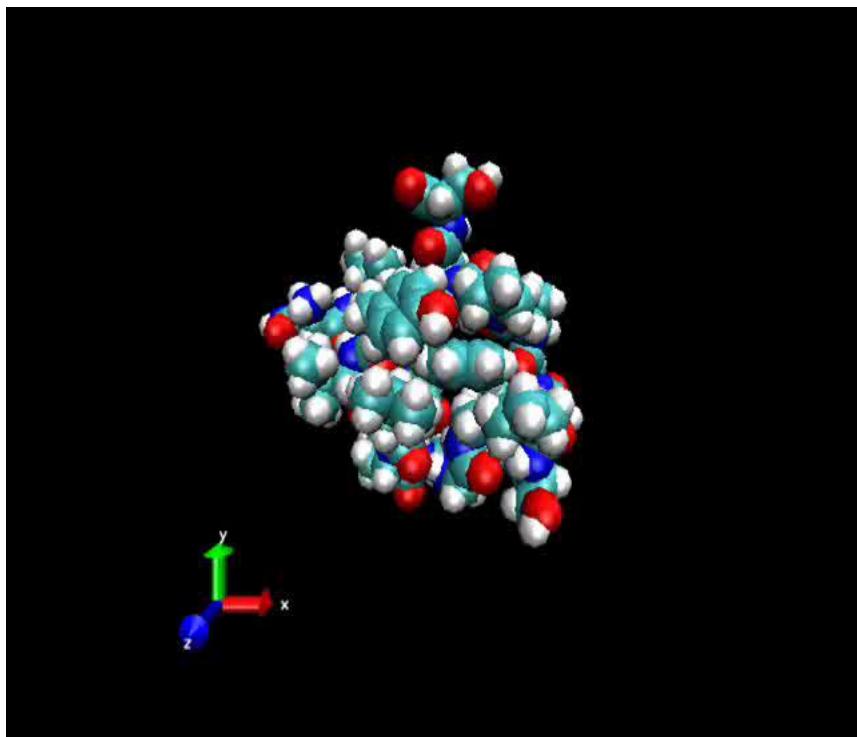


Figure 4: Electrostatic forces calculated for two separating charged atoms. The horizontal axis is the position of the moving atom in unit  $\text{\AA}$  and the vertical axis is the forces. (a) PBEQ at  $h = 0.21\text{\AA}$ ; (c) PBEQ at  $h = 0.42\text{\AA}$ ; (b) MIBPB at  $h = 0.21\text{\AA}$ ; (d) MIBPB at  $h = 0.42\text{\AA}$ .

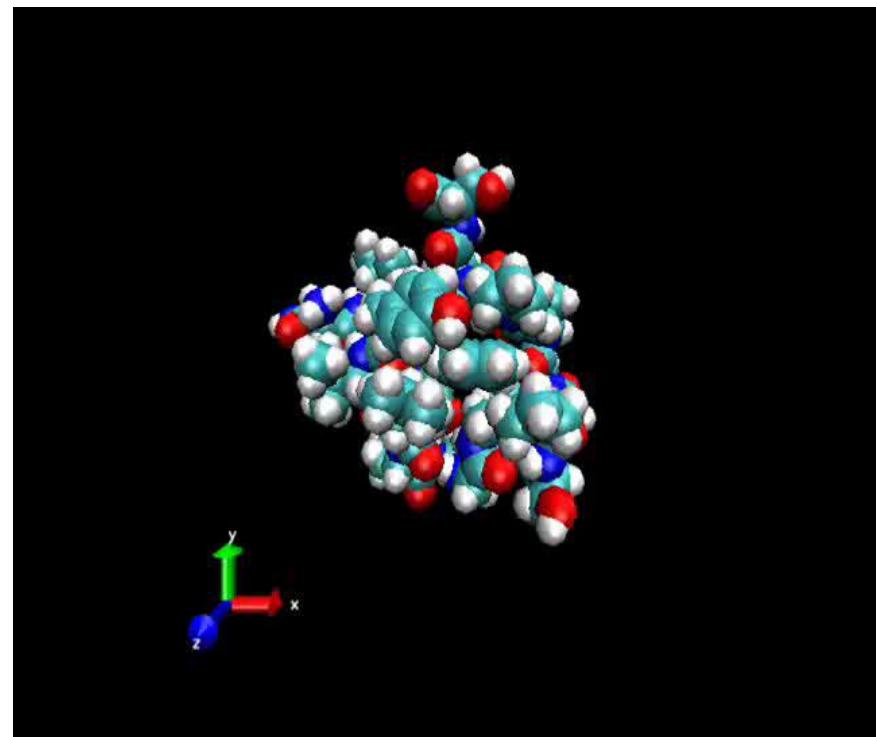
# Force convergence analysis



# Trp-cage miniprotein (1L2Y)



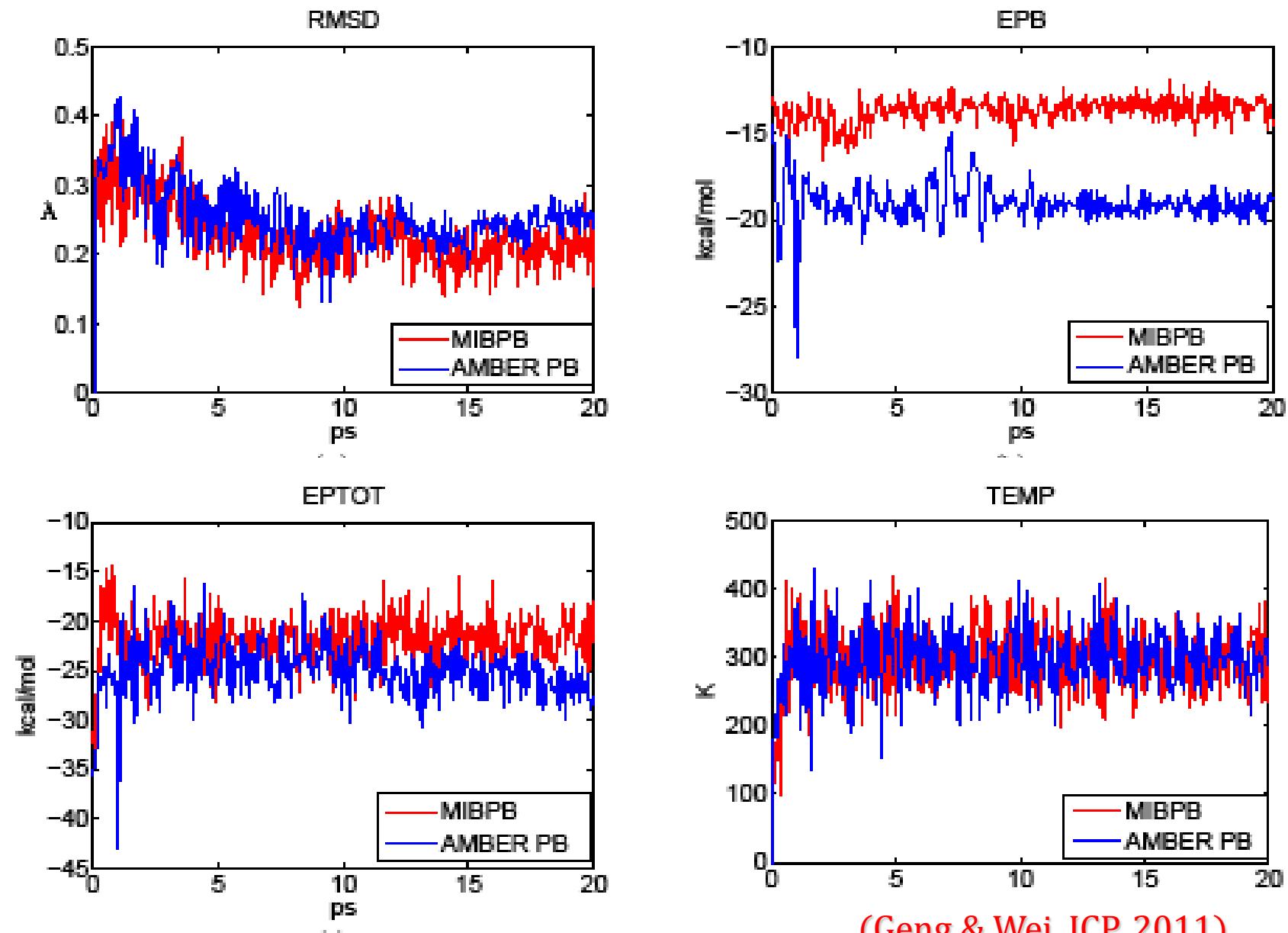
AMBER-PB



MIBPB

(Geng & Wei, JCP, 2011)

# Di-peptide



## **Further topics and future directions**

- Efficient methods for biomolecular shape analysis.
- Efficient methods for biomolecular electrostatic analysis.
- Efficient methods for multiscale dynamics.
- Efficient methods for ion transport simulation.
- Efficient methods for drug discovery simulations.
- Efficient methods for drug delivery simulations.
- Efficient methods for molecular and cellular dynamics.
- Efficient methods for macromolecular assembly.
- Efficient methods for enzymatic dynamics.
- Efficient methods for machine learning and application to biomolecules.
- Efficient methods for biomolecular Hodge decomposition.
- Discrete exterior calculus based discretizations.
- Symplectic geometry based molecular dynamics.



thank you