Two Finite Difference Methods for Poisson-Boltzmann Equation

I-Liang Chern

National Taiwan University, Taipei

Outline of this talk:

- I-L. Chern, J-L. Liu and W-C Wang, ``Accurate Evaluation for Poisson-Boltzmann Equations with Interfaces,’’ Methods and Applications of Analysis, Vol. 10, No. 2, pp. 309-328 (2003).

- Chern, I-Liang and Yu-Chen Shu, ``Coupling interface method for elliptic interface problems," Journal of Computational Physics, Vol. 225, No. 2, pp.2138-2174 (2007).

Outline of First Part:

- The problem: study the electrostatics for macromolecule in mobile ionic solution
- Numerical issue: less accuracy on electronic force due to
  - singular point charges
  - interface problems
  - open domain problem
- Treatments of singularities
  - singular charges: multipole method
  - interface problems: jump condition capturing method
  - open domain problem: coordinate partching
- Numereical examples

Outline of Second Part:

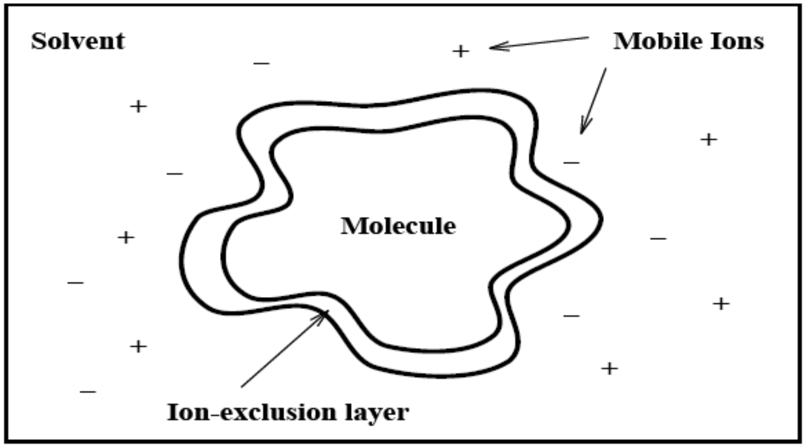
- Elliptic Interface Problems
- Coupling interface method
  - 1d: CIM1, CIM2
  - 2d: CIM2
  - d dimension
  - Hybrid CIM
  - Numerical validation
- Application to the Poisson-Boltzmann equation:
  - Treatment of complex interfaces

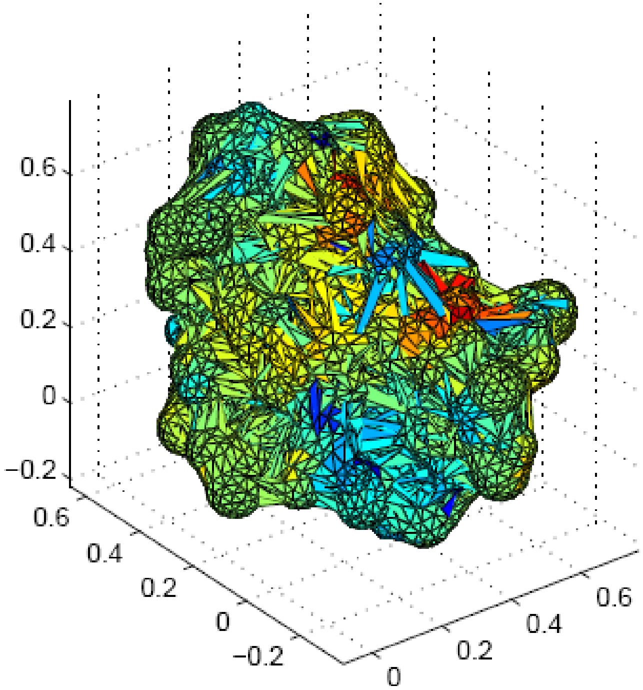
The problem

- Motivation:
  - understanding functions of macromolecules in solutions
  - drug design
- Goal : study corresponding electrostatics

The model

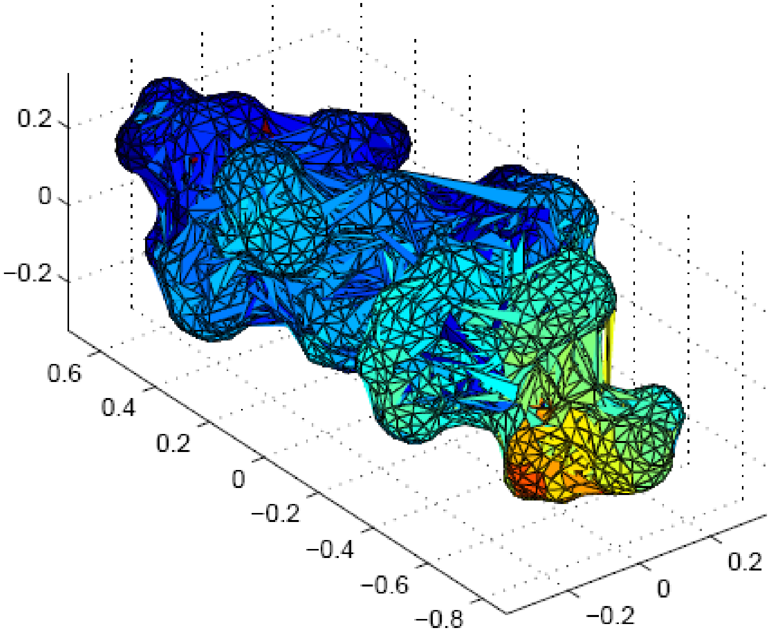
- macromolecule: a structured and polarized clusters of atoms
- ionic solvent: a continuum (Debye and Huckel 1924)





Biomolecule in solvent Poisson-Boltzmann model

- Macromolecule: 50 A
- Hydrogen layer: 1.5 – 3A
- Molecule surface: thin
- Dielectric constants:
  - 2 inside molecule
  - 80 in water



a hydrophilic protein (PDB ID:1DNG)

Equations

- **The Poisson Equation**

$$-\nabla \cdot [\epsilon(\boldsymbol{x})\nabla\Phi(\boldsymbol{x})] = K(\boldsymbol{x})(\rho_+ + \rho_-) + 4\pi Q(x)$$

- **Boltzmann distribution**

$$\begin{aligned} \rho_+ &= \left(\frac{k_B T}{e_c}\right) \exp\left(-\frac{e_c \Phi(\boldsymbol{x})}{k_B T}\right) \\ \rho_- &= \left(\frac{k_B T}{-e_c}\right) \exp\left(\frac{e_c \Phi(\boldsymbol{x})}{k_B T}\right) \end{aligned}$$

where

$$Q(x) = \sum_{i=1}^{N_m} q_i \delta(\boldsymbol{x} - \boldsymbol{x}_i)$$

$$\epsilon(\boldsymbol{x}) = \begin{cases} \epsilon_1, & \boldsymbol{x} \in \Omega_1, \\ \epsilon_2, & \boldsymbol{x} \in \Omega_2 \cup \Omega_3. \end{cases}$$

$$K(\boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} \in \Omega_1 \cup \Omega_2, \\ \bar{\kappa}^2, & \boldsymbol{x} \in \Omega_3, \end{cases}$$

$*$ $\epsilon(\boldsymbol{x})$ the dielectric parameter,

$*$ e_c the charge of an electron,

$*$ k_B the Boltzmann constant,

$*$ T the temperature,

$*$ $\boldsymbol{x}_i$ the atomic location,

$*$ q_i the atomic partial charge,

Brief Review

Previous numerical methods (from late 80 to now) for PBE can be classified into

- Finite difference methods
- Finite element methods
- Boundary element methods

- Review article: B.Z. Lu, Y.C. Zhou, M.J. Holst, J.A. McCammon, CiCP 2008

Numerical issues

less accuracy on electronic force due to

- singular point charges
- interface problems (discontinuity of dielectric function)
- open domain problem

Treatment of point charge singularities

separate singular part

$$\phi = \overline{\phi} + \tilde{\phi}.$$

where

$$\overline{\phi}(\boldsymbol{x}) = \begin{cases} \phi^*(\boldsymbol{x}) + \phi^0(\boldsymbol{x}) & \boldsymbol{x} \in \Omega_1 \\ 0 & \boldsymbol{x} \in \Omega_2 \cup \Omega_3 \end{cases}.$$

and ϕ^* is the potential in the free space induced by Q, i.e.

$$\phi^*(\boldsymbol{x}) = \begin{cases} C \sum_{i=1}^{m} \frac{1}{\epsilon_1} \frac{z_i}{4\pi} \frac{1}{|\boldsymbol{x} - \boldsymbol{x}_i|}, & \boldsymbol{x} \in R^3 \\ C \sum_{i=1}^{m} -\frac{1}{\epsilon_1} \frac{z_i}{2\pi} \log(|\boldsymbol{x} - \boldsymbol{x}_i|), & \boldsymbol{x} \in R^2 \end{cases}.$$

ϕ^0 is a harmonic function in Ω_1 satisfying

$$\begin{cases} \triangle \phi^0 = 0 & \text{in } \Omega_1 \\ \phi^0 = -\phi^* & \text{on } \Gamma_1. \end{cases}$$

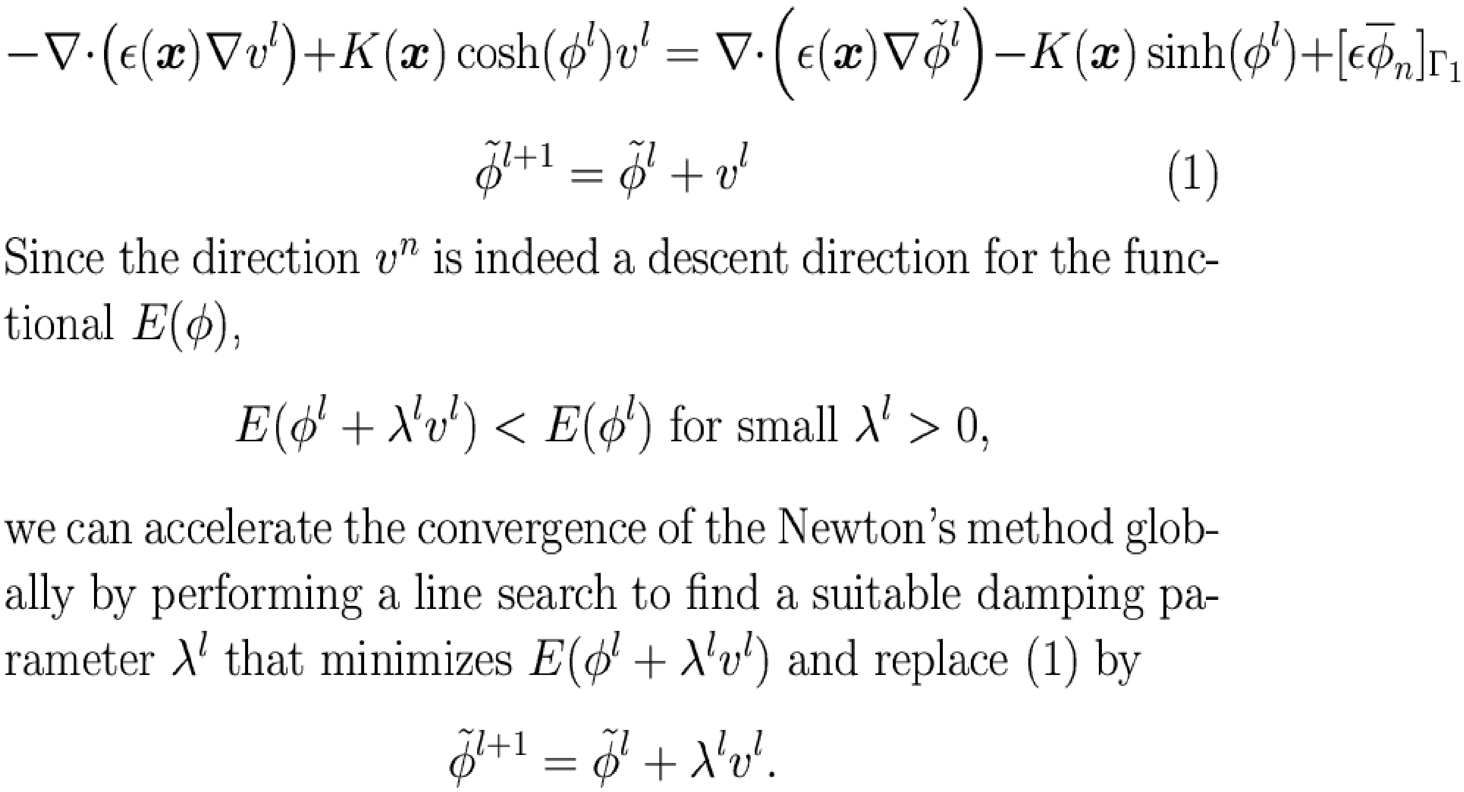
The introduction of ϕ^0 is to force $[\overline{\phi}] = 0$ across Γ_1.

The correction potential satisfies

$$-\nabla \cdot \left(\epsilon(\boldsymbol{x}) \nabla \tilde{\phi}(\boldsymbol{x}) \right) + K(\boldsymbol{x}) \sinh(\tilde{\phi}(\boldsymbol{x})) = [\epsilon \overline{\phi}_n]_{\Gamma_1} \delta_{\Gamma_1}.$$

Thus, the point charge singularity is transfered into surface singularity.

Damped Newton’s Method



$$-\nabla\cdot\left(\epsilon(\boldsymbol{x})\nabla v^l\right)+K(\boldsymbol{x})\cosh(\phi^l)v^l = \nabla\cdot\left(\epsilon(\boldsymbol{x})\nabla\tilde{\phi}^l\right)-K(\boldsymbol{x})\sinh(\phi^l)+[\epsilon\overline{\phi}_n]_{\Gamma_1}$$

$$\tilde{\phi}^{l+1} = \tilde{\phi}^l + v^l \tag{1}$$

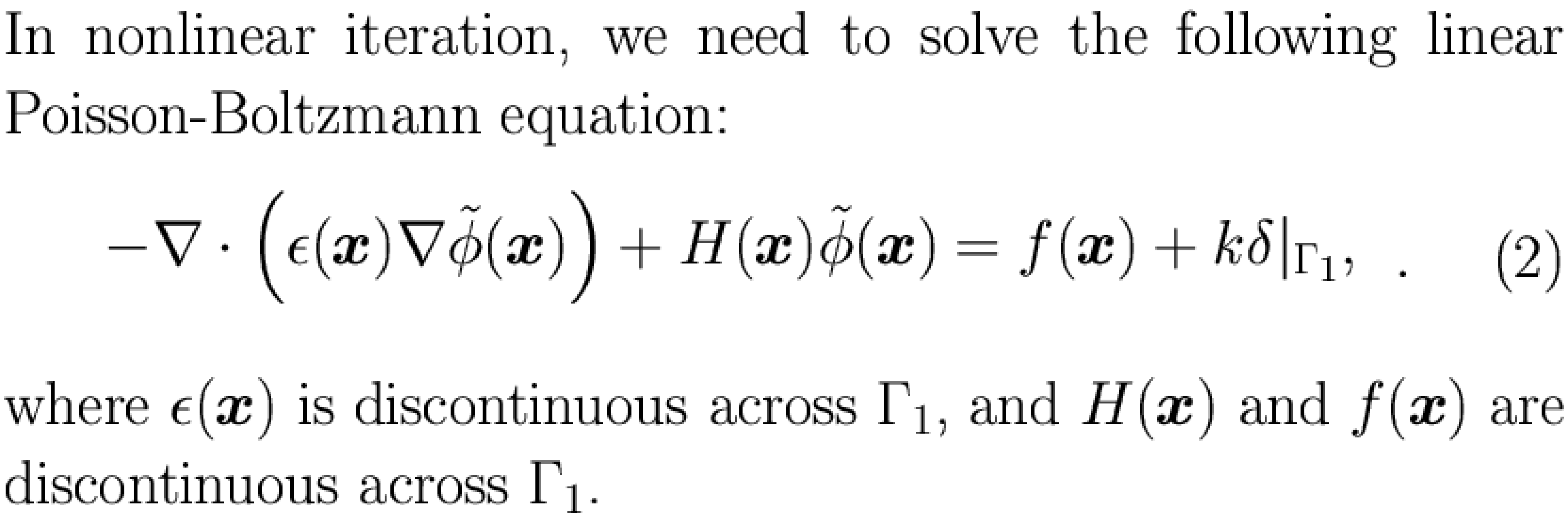
Since the direction v^n is indeed a descent direction for the functional $E(\phi)$,

$$E(\phi^l + \lambda^l v^l) < E(\phi^l) \text{ for small } \lambda^l > 0,$$

we can accelerate the convergence of the Newton’s method globally by performing a line search to find a suitable damping parameter λ^l that minimizes $E(\phi^l + \lambda^l v^l)$ and replace (1) by

$$\tilde{\phi}^{l+1} = \tilde{\phi}^l + \lambda^l v^l.$$

Treatment of surface singularities



In nonlinear iteration, we need to solve the following linear Poisson-Boltzmann equation:

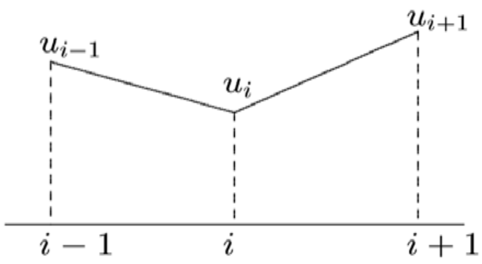
$$-\nabla\cdot\Big(\epsilon(\boldsymbol{x})\nabla\tilde{\phi}(\boldsymbol{x})\Big)+H(\boldsymbol{x})\tilde{\phi}(\boldsymbol{x})=f(\boldsymbol{x})+k\delta|_{\Gamma_1}, \quad . \tag{2}$$

where $\epsilon(\boldsymbol{x})$ is discontinuous across Γ_1, and $H(\boldsymbol{x})$ and $f(\boldsymbol{x})$ are discontinuous across Γ_1.

- **the jump condition capturing scheme**
- **skew variable**

Jump condition capturing scheme

- **1-D Case**



- uniform grid
- interface is on the grid
- The 1-d equation with singular source

$$(\epsilon(x)u')' = f + k\delta(x - x_i)$$

  is discretized by

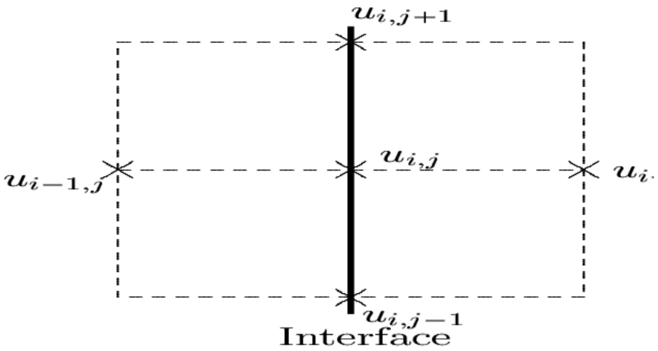
$$\frac{1}{\Delta x}\left(\epsilon_{i+1/2}\left(\frac{u_{i+1}-u_i}{\Delta x}\right) - \epsilon_{i-1/2}\left(\frac{u_i-u_{i-1}}{\Delta x}\right)\right)$$
$$= \frac{1}{2}(f_{i,+} + f_{i,-}) + \frac{k}{\Delta x}$$

- truncation error
  - $\ast$ $O(\Delta x^2)$ off the interface
  - $\ast$ $O(\Delta x)$ on the interface
- global error $O(\Delta x^2)$

Jump condition capturing scheme

- **2-D Case: (A wrong approach)**

$$\frac{1}{\Delta x}\left(\epsilon_{i+1/2,j}\left(\frac{u_{i+1,j}-u_{i,j}}{\Delta x}\right)-\epsilon_{i-1/2,j}\left(\frac{u_{i,j}-u_{i-1,j}}{\Delta x}\right)\right)$$
$$+\frac{1}{\Delta y}\left(\bar{\epsilon}_{i,j+1/2}\left(\frac{u_{i,j+1}-u_{i,j}}{\Delta y}\right)-\bar{\epsilon}_{i,j-1/2}\left(\frac{u_{i,j}-u_{i,j-1}}{\Delta x}\right)\right)$$
$$=\frac{1}{2}(f_{i+,j}+f_{i-,j})+\frac{k}{\Delta x}$$



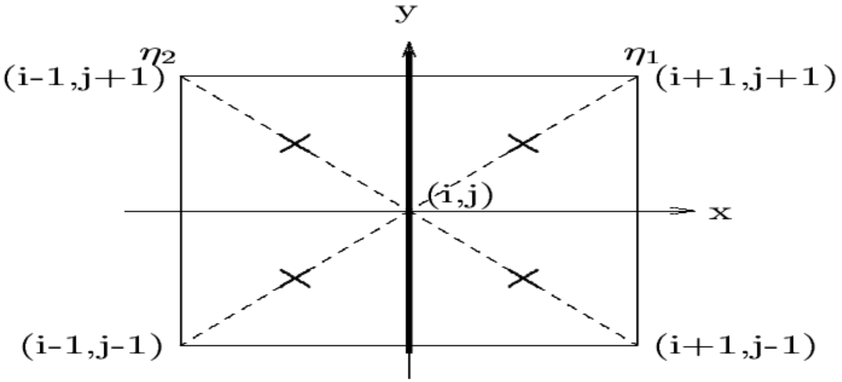
– $\bar{\epsilon}_{i,j+1/2}$ is not defined.

– The local truncation error is $O(1)$.

Jump condition capturing scheme

- **2-D Case: (A right approach) --- Use skew variable**

$$
\begin{aligned}
&\frac{1}{\Delta\eta_1}\left(\epsilon_{i+1/2,j+1/2}\left(\frac{u_{i+1,j+1}-u_{i,j}}{\Delta\eta_1}\right)-\epsilon_{i-1/2,j-1/2}\left(\frac{u_{i,j}-u_{i-1,j-1}}{\Delta\eta_1}\right)\right)\\
&+\frac{1}{\Delta\eta_2}\left(\epsilon_{i-1/2,j+1/2}\left(\frac{u_{i-1,j+1}-u_{i,j}}{\Delta\eta_2}\right)-\epsilon_{i+1/2,j-1/2}\left(\frac{u_{i,j}-u_{i+1,j-1}}{\Delta\eta_2}\right)\right)\\
&=\frac{1}{2}(f_{i+,j}+f_{i-,j})+k\left(\frac{1}{\Delta\eta_1}+\frac{1}{\Delta\eta_2}\right)
\end{aligned}
$$

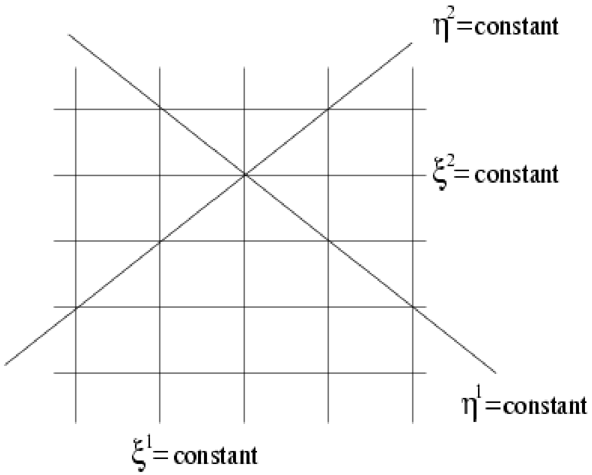


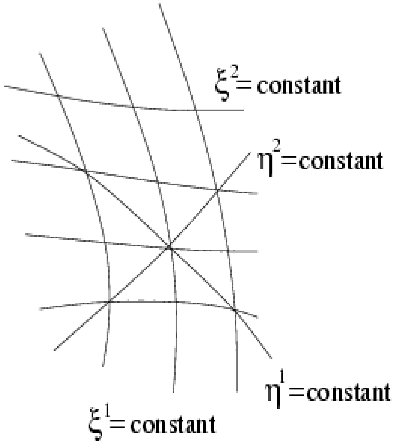
$$\frac{\partial}{\partial\eta_1}\left(\epsilon\frac{\partial u}{\partial\eta_1}\right)+\frac{\partial}{\partial\eta_2}\left(\epsilon\frac{\partial u}{\partial\eta_2}\right)=f+k\delta|_{\Gamma_1}$$

General Case

Let us denote by (ξ^1, ξ^2) the variables in the computational domain, $\boldsymbol{X}(\xi^1, \xi^2) \in R^2$ the position vector in the physical space.

we define





$$\eta^1 = \frac{\xi^1 \Delta\xi^2 + \xi^2 \Delta\xi^1}{\sqrt{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}}$$

$$\eta^2 = \frac{\xi^2 \Delta\xi^1 - \xi^1 \Delta\xi^2}{\sqrt{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}}$$

$$\Delta\eta^1 = \Delta\eta^2 = \frac{2\Delta\xi^1 \Delta\xi^2}{\sqrt{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}}$$

finite difference discretization in the skewed variable

$$\boxed{-\partial_\mu(\epsilon\sqrt{\hat{g}}\hat{g}^{\mu\nu}\partial_\nu\tilde{\phi}) + \sqrt{\hat{g}}H(x)\tilde{\phi} = \sqrt{\hat{g}}f}$$

$$\hat{g}^{\mu\nu} = \langle\nabla\eta^\mu, \nabla\eta^\nu\rangle$$

$$\hat{g}_{\mu\nu} = \langle\frac{\partial\boldsymbol{X}}{\partial\eta^\mu}, \frac{\partial\boldsymbol{X}}{\partial\eta^\nu}\rangle$$

$$\hat{g} = \det(\hat{g}_{\mu\nu})$$

$$\sqrt{\hat{g}} = \det(\frac{\partial\boldsymbol{X}}{\partial\boldsymbol{\eta}}) = \frac{1}{2}\left(\frac{\Delta\xi^1}{\Delta\xi^2} + \frac{\Delta\xi^2}{\Delta\xi^1}\right)\det(\frac{\partial\boldsymbol{X}}{\partial\boldsymbol{\xi}})$$

$$\hat{g}^{\mu\gamma}\hat{g}_{\gamma\nu} = \delta^\mu_\nu$$

finite difference discretization in the skewed variable

the jump conditions can be incorporated into the following finite difference discretization

$$-\frac{1}{(\Delta\eta)^2}\left(\begin{array}{l}(\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{11})_{i+\frac{1}{2},j+\frac{1}{2}}(\tilde{\phi}_{i+1,j+1}-\tilde{\phi}_{i,j})-(\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{11})_{i-\frac{1}{2},j-\frac{1}{2}}(\tilde{\phi}_{i,j}-\tilde{\phi}_{i-1,j-1})\\ +(\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{12})_{i+\frac{1}{2},j+\frac{1}{2}}(\tilde{\phi}_{i,j+1}-\tilde{\phi}_{i+1,j})-(\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{12})_{i-\frac{1}{2},j-\frac{1}{2}}(\tilde{\phi}_{i-1,j}-\tilde{\phi}_{i,j-1})\\ +(\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{21})_{i-\frac{1}{2},j+\frac{1}{2}}(\tilde{\phi}_{i,j+1}-\tilde{\phi}_{i-1,j})-(\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{21})_{i+\frac{1}{2},j-\frac{1}{2}}(\tilde{\phi}_{i+1,j}-\tilde{\phi}_{i,j-1})\\ +(\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{22})_{i-\frac{1}{2},j+\frac{1}{2}}(\tilde{\phi}_{i-1,j+1}-\tilde{\phi}_{i,j})-(\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{22})_{i+\frac{1}{2},j-\frac{1}{2}}(\tilde{\phi}_{i,j}-\tilde{\phi}_{i+1,j-1})\end{array}\right)$$

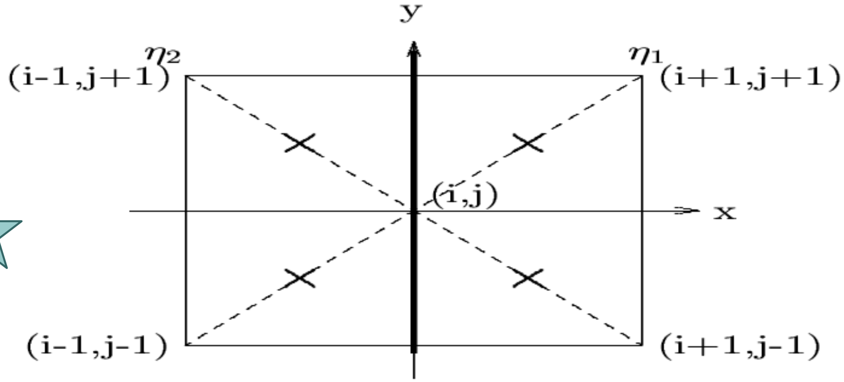
$$=\sqrt{(\hat{g}_h)_{i,j}}(f_{i,j}-H_{i,j}\tilde{\phi}_{i,j})$$

At the interface, the right-hand side is replaced by

$$\sqrt{(\hat{g}_h)_{i,j}}\left(\frac{1}{2}(f_{i^+,j}+f_{i^-,j})-\frac{1}{2}(H_{i^+,j}\tilde{\phi}_{i^+,j}+H_{i^-,j}\tilde{\phi}_{i^-,j})\right)$$

$$+\text{ terms involving }[\tilde{\phi}]\text{ and }[\epsilon\tilde{\phi}_n]$$

finite difference discretization in the skewed variable



$$((\hat{g}_h)_{11})_{i+\frac{1}{2},j+\frac{1}{2}}$$
$$\langle D_1\boldsymbol{X}_{i+\frac{1}{2},j+\frac{1}{2}}, D_1\boldsymbol{X}_{i+\frac{1}{2},j+\frac{1}{2}}\rangle$$
$$\langle \frac{\boldsymbol{X}_{i+1,j+1}-\boldsymbol{X}_{i,j}}{\Delta\eta}, \frac{\boldsymbol{X}_{i+1,j+1}-\boldsymbol{X}_{i,j}}{\Delta\eta}\rangle$$

- The local truncation error is $O(\Delta\eta)$ on the interface and $O(\Delta\eta^2)$ elsewhere.
- The resulting flux function is 2nd order accurate, even on the interface.
- The symmetry and positivity of this discretization is essentially unconditional.

Summary of the Algorithm

- **Step 1** Evaluate the free space Poisson kernel $\phi^*(\boldsymbol{x})$ and $\phi^*_n(\boldsymbol{x})$ for $\boldsymbol{x} \in \Gamma_1$ (by multipole method).
- **Step 2** Evaluate $\phi^0_n(x)$ for $x \in \Gamma_1$ by solving the Laplace equation in Ω_1 so that $\tilde{\phi} = 0$ on Γ_1.
- **Step 3** Compute the correction potential $\tilde{\phi}$ by the damped Newton's method. The resulting linearized Poisson Boltzmann equation is discretized using the jump condition capturing scheme. The initial trial of $\tilde{\phi}$ of the iteration is set to be zero.

Numerical experiments

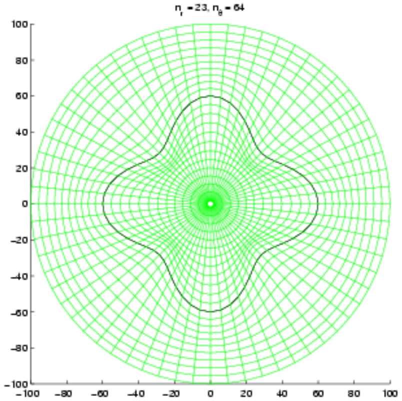


Figure 5: The shape of the macromolecule for Example 1

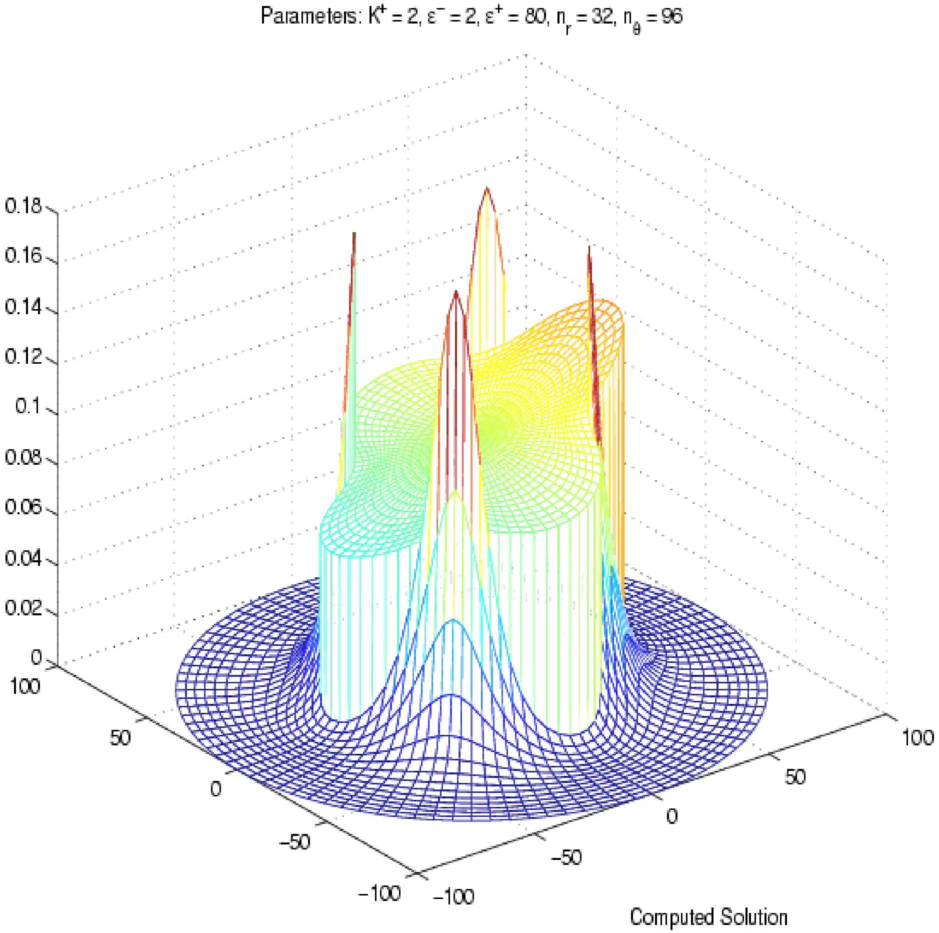
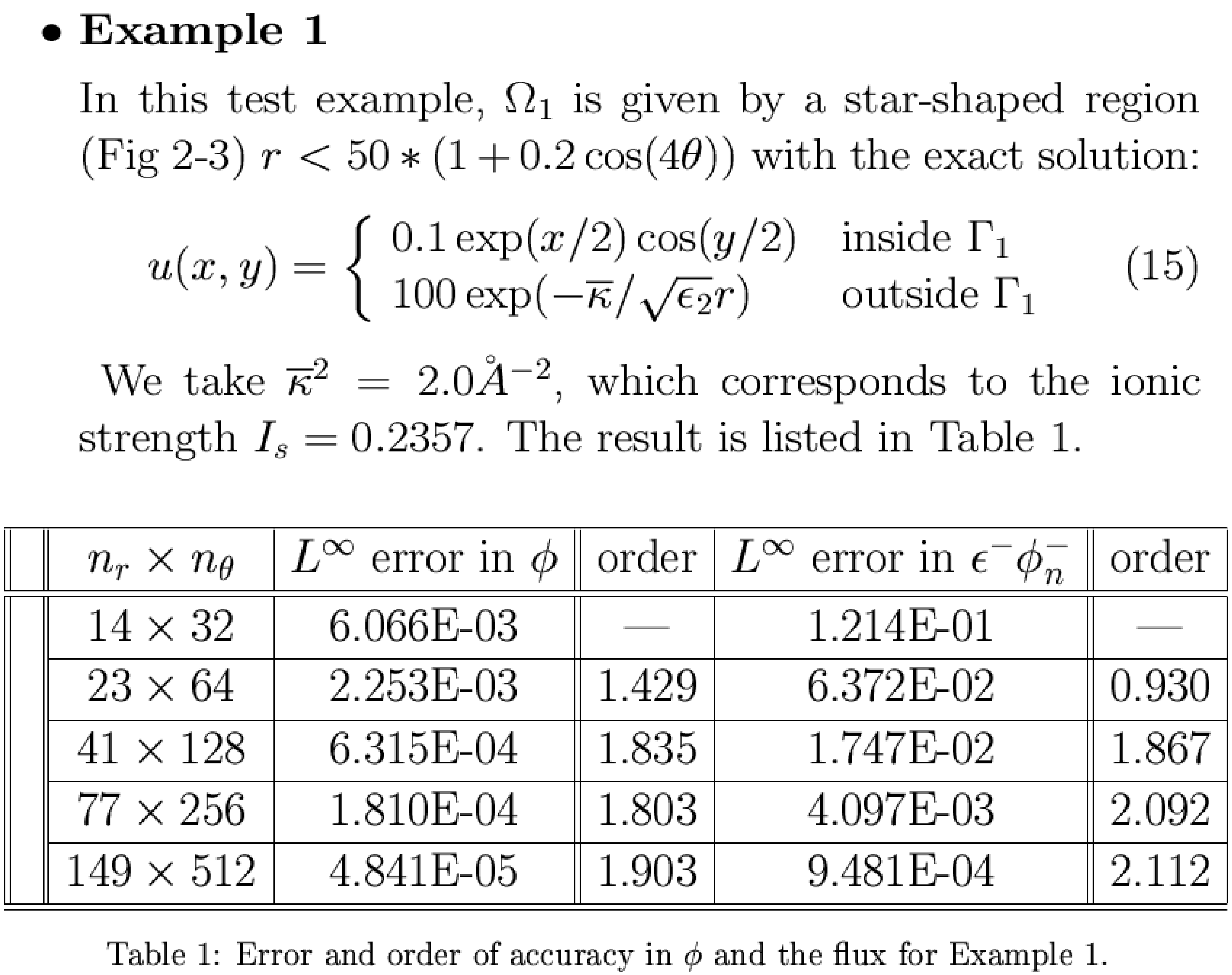


Figure 6: Computed solution for Example 1

Numerical experiments



- **Example 1**

  In this test example, Ω_1 is given by a star-shaped region (Fig 2-3) $r < 50 * (1 + 0.2\cos(4\theta))$ with the exact solution:

$$u(x,y) = \begin{cases} 0.1\exp(x/2)\cos(y/2) & \text{inside } \Gamma_1 \\ 100\exp(-\overline{\kappa}/\sqrt{\epsilon_2}r) & \text{outside } \Gamma_1 \end{cases} \tag{15}$$

We take $\overline{\kappa}^2 = 2.0\mathring{A}^{-2}$, which corresponds to the ionic strength $I_s = 0.2357$. The result is listed in Table 1.

	$n_r \times n_\theta$	L^∞ error in ϕ	order	L^∞ error in $\epsilon^-\phi_n^-$	order
	14×32	6.066E-03	—	1.214E-01	—
	23×64	2.253E-03	1.429	6.372E-02	0.930
	41×128	6.315E-04	1.835	1.747E-02	1.867
	77×256	1.810E-04	1.803	4.097E-03	2.092
	149×512	4.841E-05	1.903	9.481E-04	2.112

Table 1: Error and order of accuracy in ϕ and the flux for Example 1.

Numerical experiments

- **Example 4**

  In this example, we perform an actual simulation on the Poisson-Boltzmann equation. The interface is the same as in Example 3. Here we take $\overline{\kappa}^2 = 1.27 \mathring{A}^{-2}$ and $C = 15,000$. In case (a), we put six charges with alternating sign corresponding to $z_i = \pm 1$. In case (b), z_i's are randomly chosen between ± 1 and sum to zero. The results are plotted in Fig 9-11.

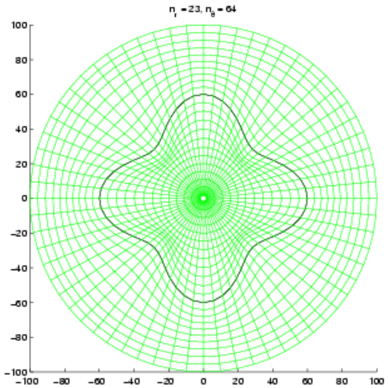


Figure 5: The shape of the macromolecule for Example 1

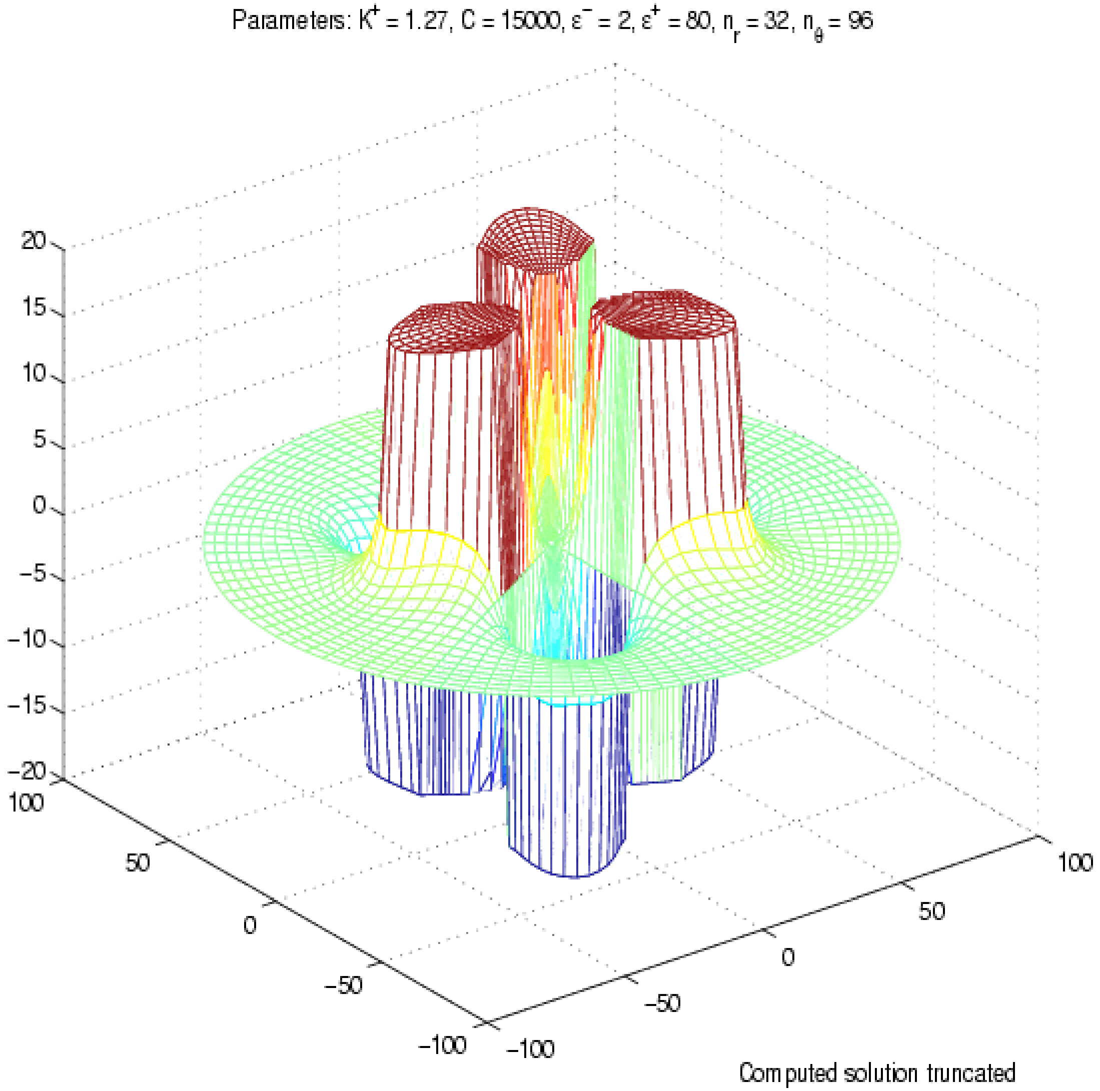


Figure 12: Computed solution for Example 4a

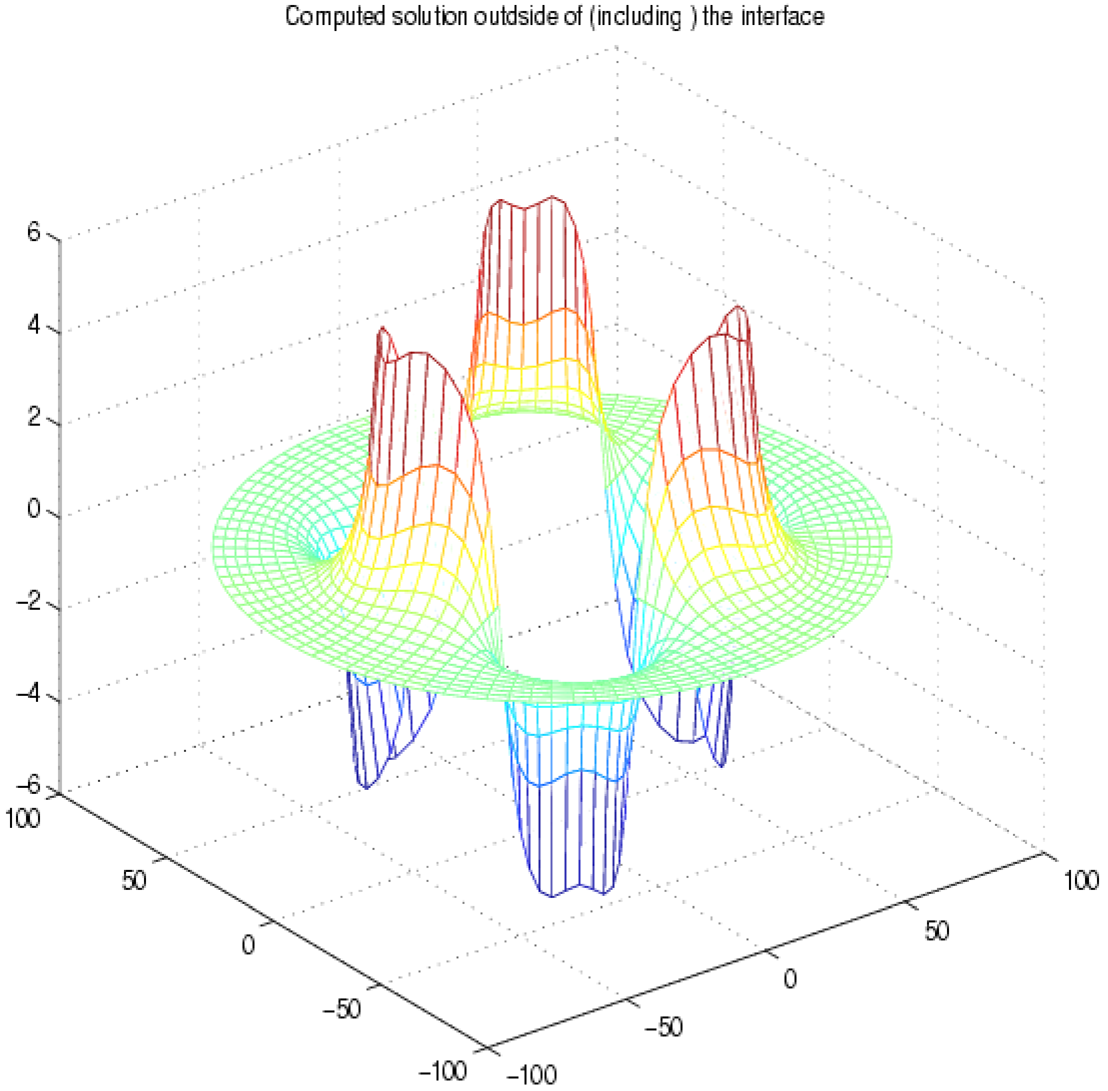


Figure 13: Computed solution outside of Γ_1 for Example 4a

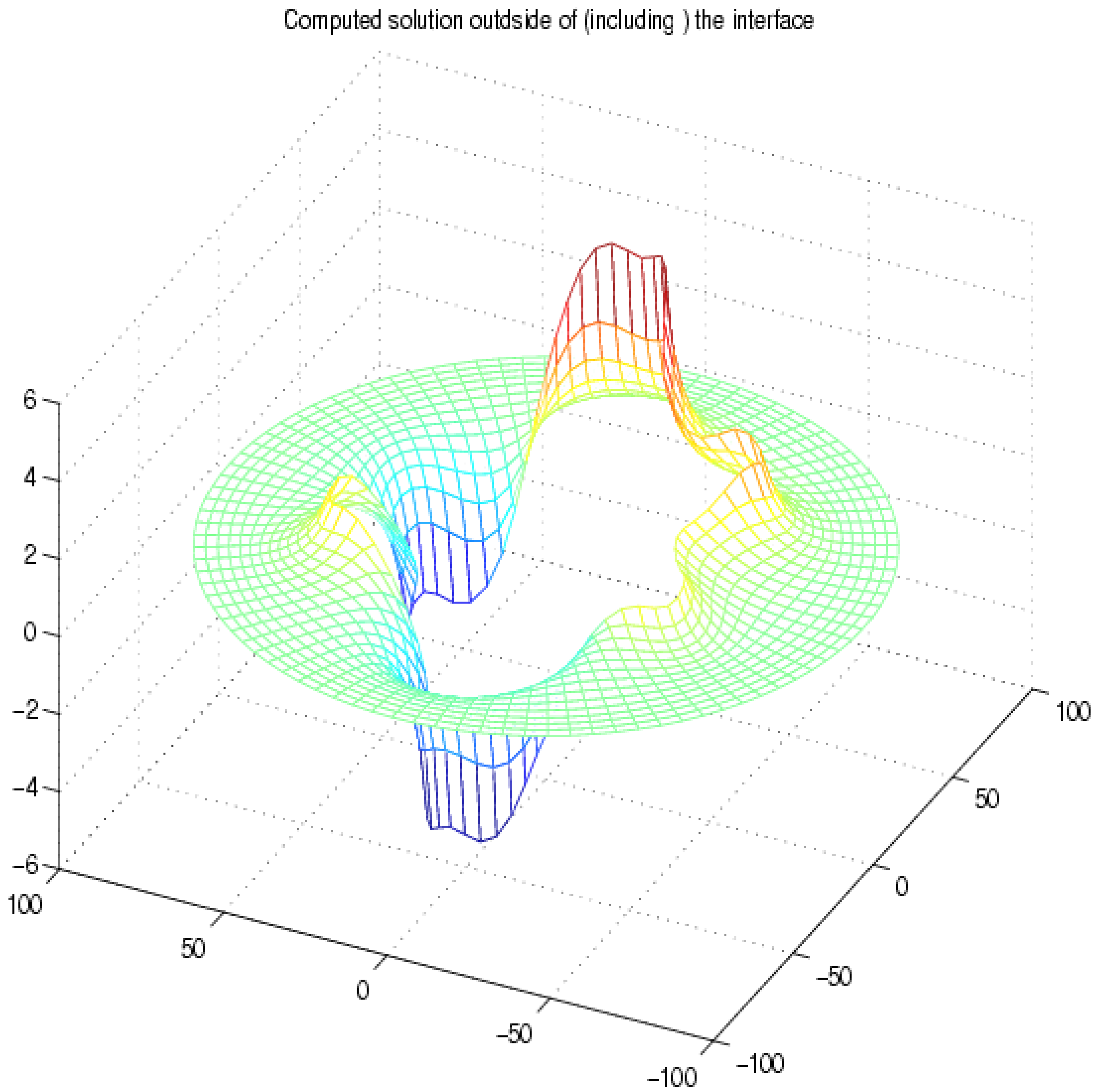


Figure 15: Computed solution outside of Γ_1 for Example 4b

Part 1-Conclusion

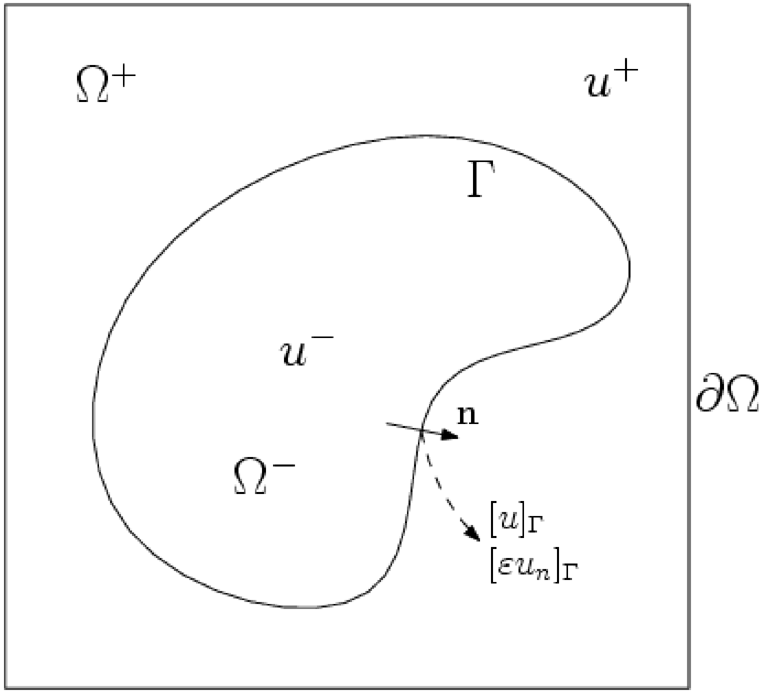
1. Point charge singularity is resolved by Green’s function and a harmonic function
2. Surface singularity is solved by jump condition capturing method with a skew variable
3. The resulting linear system is symmetric and positive definite, standard fast solver can be adopted
4. Second order accurate for electric field

Part 2: Coupling Interface Method for Elliptic Interface Problems

$$-\nabla\cdot(\varepsilon(\mathbf{x})\nabla u(\mathbf{x})) = f(\mathbf{x}),\ \ \mathbf{x}\in\Omega\backslash\Gamma,$$

$$[u]=\tau,\ [\varepsilon u_n]=\sigma \quad \text{on } \Gamma,$$

$$u = g \quad \text{on } \partial\Omega.$$



ε and *u* are discontinuous,

f is singular across Γ

Three classes of approaches

- Boundary integral approach

- Finite element approach:

- Finite Difference approach:
  - Body-fitting approach
  - Fixed underlying grid: more flexible for moving interface problems

Regular Grid Methods for Solving Elliptic Interface Problems

- **Regularization approach** (Tornberg-Engquist, 2003)
  - □ Harmonic Averaging (Tikhonov-Samarskii, 1962)
  - □ Immersed Boundary Method (IB Method) (Peskin, 1974)
  - □ Phase field method

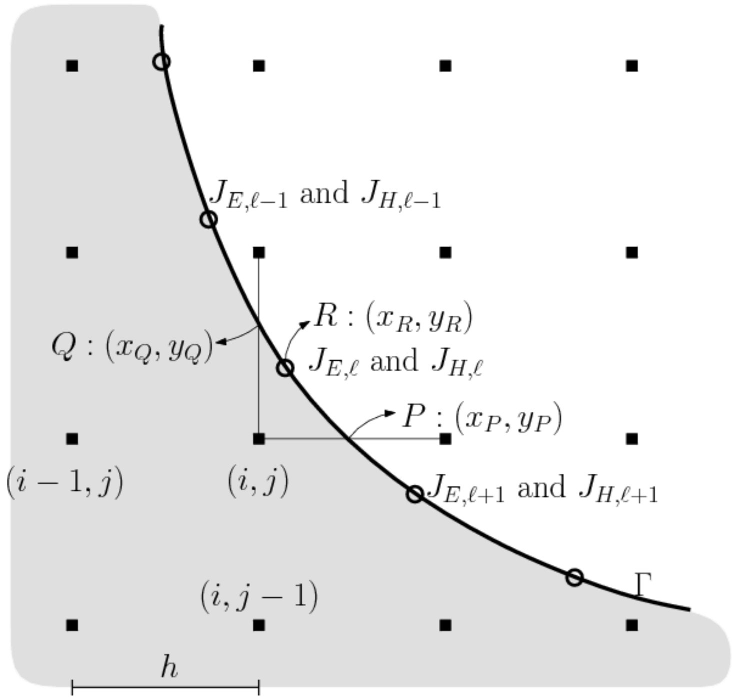
- **Dimension un-splitting approach**
  - □ Immersed Interface Method (IIM) (LeVeque-Li, 1994)
  - □ Maximum Principle Preserving IIM (MIIM) (Li-Ito, 2001)
  - □ Fast iterative IIM (FIIM) (Li, 1998)

- **Dimension splitting approach**
  - □ Ghost Fluid Method (Fedkiw et al., 1999, Liu et al. 2000)
  - □ Decomposed Immersed Interface Method (DIIM) (Berthelsen, 2004)
  - □ Matched Interface and Boundary Method (MIB) (YC Zhou et al., 2006)
  - □ Coupling interface method (CIM) (Chern and Shu 2007)

Coupling Interface Method (CIM)

- CIM
    - CIM1 (first order)
    - CIM2 (2nd order)
    - Hybrid CIM (CIM1 + CIM2)

      for complex interface problems
- Augmented CIM
    - Auxiliary variables on interfaces


$J_{E,\ell-1}$ and $J_{H,\ell-1}$
$R:(x_R,y_R)$
$Q:(x_Q,y_Q)$
$J_{E,\ell}$ and $J_{H,\ell}$
$P:(x_P,y_P)$
$(i-1,j)$
(i,j)
$J_{E,\ell+1}$ and $J_{H,\ell+1}$
Γ
$(i,j-1)$
h

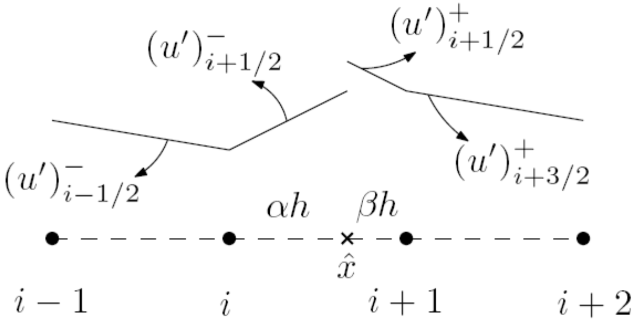
Numerical Issues for dealing with interface problems

- Accuracy: second-order in maximum norm.
- Simplicity: easy to derive and program.
- Stability: nice stencil coefficients for linear solvers.
- Robustness: capable to handle complex interfaces.

CIM outline

- 1d: CIM1, CIM2
- 2d: CIM2
- d dimension
- Hybrid CIM
- Numerical validation
- Application to the Poisson-Boltzmann equation

CIM1: one dimension



$$\begin{cases} u^-(x) := u_i + (u')^-_{i+1/2}(x - x_i) & \text{for } x_i \le x < \hat{x} \\ u^+(x) := u_{i+1} + (u')^+_{i+1/2}(x - x_{i+1}) & \text{for } \hat{x} < x < x_{i+1}. \end{cases}$$

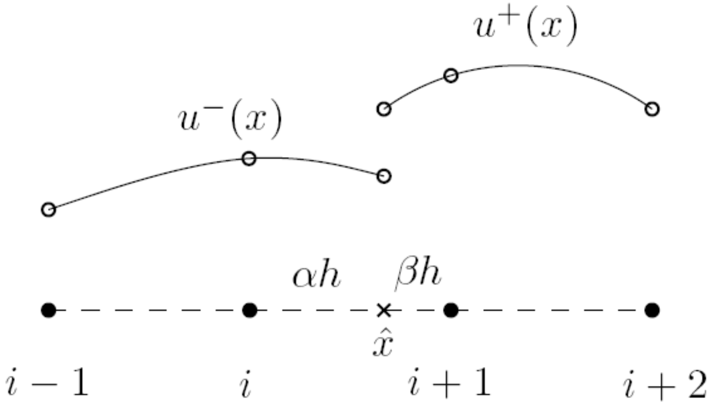
$$\begin{aligned} (u_{i+1} - \beta h (u')^+_{i+1/2}) - (u_i + \alpha h (u')^-_{i+1/2}) &\approx \tau \\ \varepsilon^+ (u')^+_{i+1/2} - \varepsilon^- (u')^-_{i+1/2} &\approx \sigma. \end{aligned}$$

$$\begin{aligned} (u')^-_{i+1/2} &= \frac{1}{h}\left(\bar{\rho}^+(u_{i+1} - u_i) - \bar{\rho}^+\tau - \beta h \frac{\sigma}{\bar{\varepsilon}}\right) + O(h) \\ (u')^+_{i+1/2} &= \frac{1}{h}\left(\bar{\rho}^-(u_{i+1} - u_i) - \bar{\rho}^-\tau + \alpha h \frac{\sigma}{\bar{\varepsilon}}\right) + O(h) \end{aligned}$$

$$\bar{\varepsilon} = \alpha\varepsilon^+ + \beta\varepsilon^-, \; \bar{\rho}^\pm = \varepsilon^\pm / \bar{\varepsilon}.$$

$$-(\varepsilon u')'(x_i) = -\frac{1}{h}\varepsilon_i \left((u')^-_{i+1/2} - (u')^-_{i-1/2}\right) + O(1).$$

CIM2: One dimension



- Quadratic approximation and match two grid

$$
\begin{aligned}
u_\ell(x) &= u_i + \left(\frac{u_i - u_{i-1}}{h} + \frac{1}{2}\underline{hu_i''}\right)(x - x_i) + \frac{1}{2}\underline{u_i''}(x - x_i)^2 + O(h^3),\\
u_r(x) &= u_{i+1} + \left(\frac{u_{i+2} - u_{i+1}}{h} - \frac{1}{2}\underline{hu_{i+1}''}\right)(x - x_{i+1}) + \frac{1}{2}\underline{u_{i+1}''}(x - x_{i+1})^2 + O(h^3)
\end{aligned}
$$

- Match two jump conditions

$$
u_r(\hat{x}) - u_\ell(\hat{x}) = \tau, \quad \hat{\varepsilon}^+ u_r'(\hat{x}) - \hat{\varepsilon}^- u_\ell'(\hat{x}) = \sigma,
$$

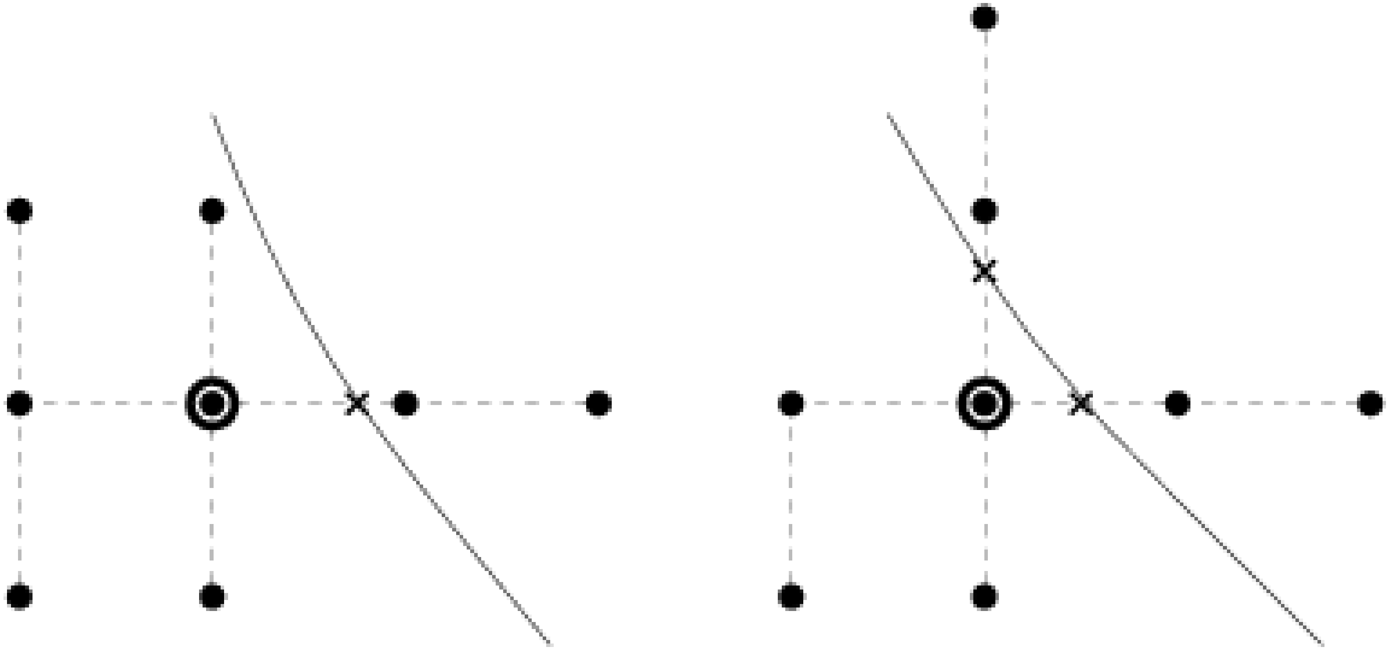
CIM2: One dimension

$$
\begin{aligned}
u_i'' &= \frac{1}{h^2}\left(L^{(\ell)}u_i + J_i^{(\ell)}\right) + O(h) \\
u_{i+1}'' &= \frac{1}{h^2}\left(L^{(r)}u_{i+1} + J_{i+1}^{(r)}\right) + O(h),
\end{aligned}
$$

$$
\begin{aligned}
L^{(\ell)}u_i &:= a_{i,-1}u_{i-1} + a_{i,0}u_i + a_{i,1}u_{i+1} + a_{i,2}u_{i+2} \\
L^{(r)}u_{i+1} &:= a_{i+1,-2}u_{i-1} + a_{i+1,-1}u_i + a_{i+1,0}u_{i+1} + a_{i+1,1}u_{i+2} \\
J_i^{(\ell)} &:= a_{i,\tau}\frac{\tau}{h^2} + a_{i,\sigma}\frac{\sigma}{\hat{\varepsilon}h} \\
J_{i+1}^{(r)} &:= -a_{i+1,\tau}\frac{\tau}{h^2} + a_{i+1,\sigma}\frac{\sigma}{\hat{\varepsilon}h}.
\end{aligned}
$$

CIM2: 2 dimensions

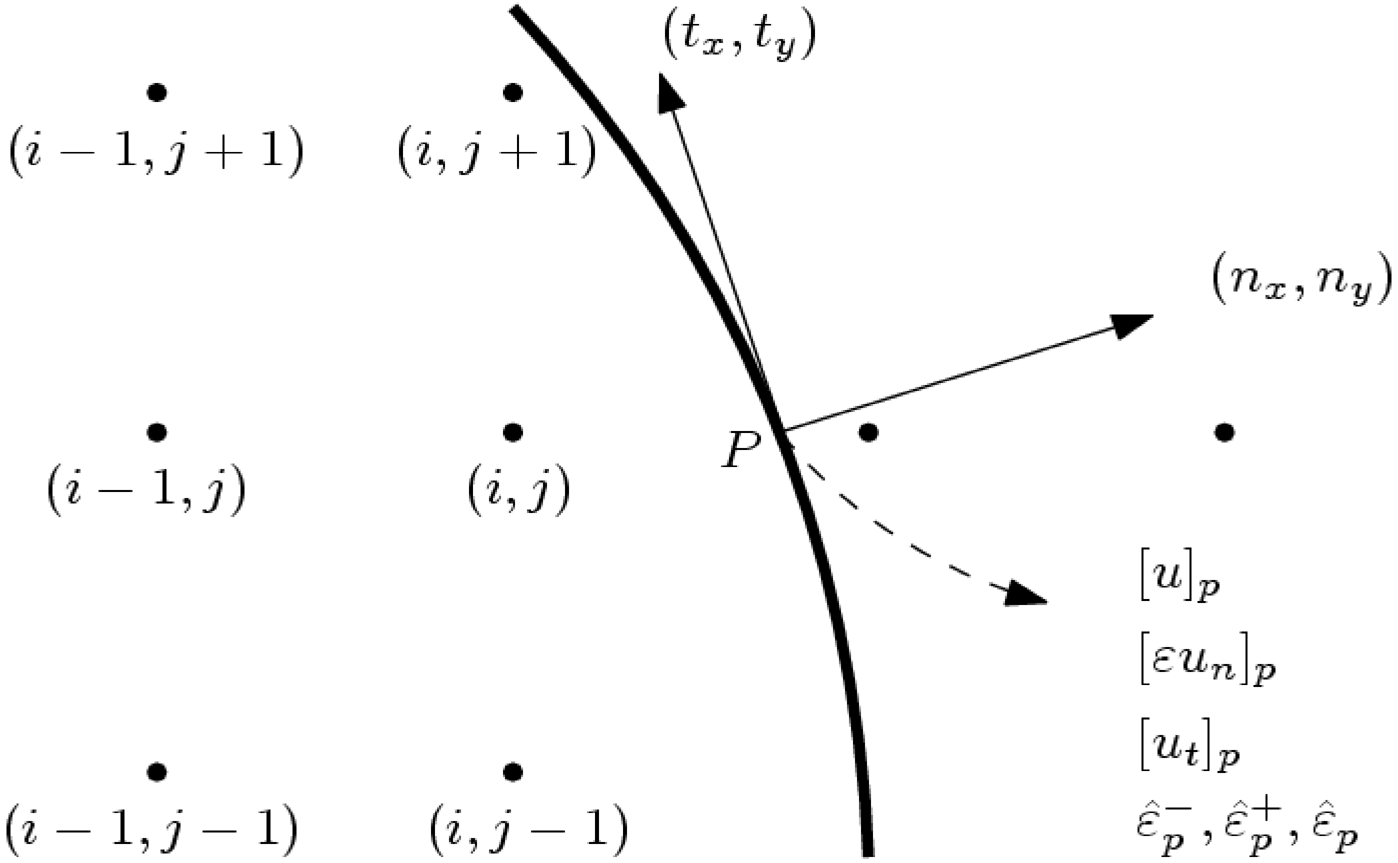
Stencil at a normal on-front points (bullet) (8 points stencil)



(a) Two dimension: 2 cases

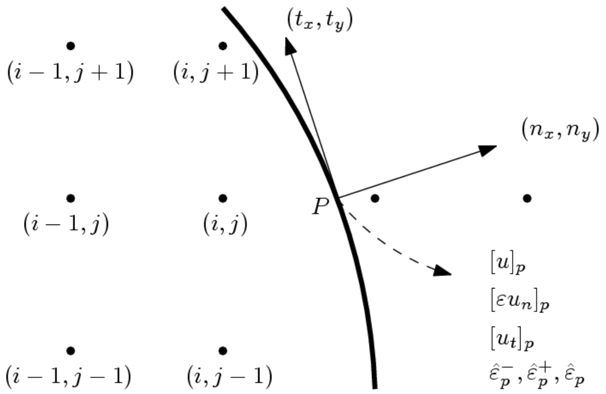
Coupling Interface Method

CIM2 Case 1:



Coupling Interface Method

CIM2 (Case 1):



- Dimension splitting approach

$$\underline{u_{xx}} = \frac{1}{h^2}\left(Lu + a_\tau [u]_p + a_\sigma h \frac{[\varepsilon u_x]_p}{\hat{\varepsilon}_p}\right) + O(h)$$

- Decomposition of jump condition

$$[\varepsilon u_x]_p = [\varepsilon u_n]_p n_x + (\hat{\varepsilon}_p^+ [u_t]_p + (\hat{\varepsilon}_p^+ - \hat{\varepsilon}_p^-)(u_t^-)_p)\,(t_x)$$

- One side interpolation

$$\begin{aligned}(u_t^-)_p &= \left(\frac{u_{i,j} - u_{i-1,j}}{h} + (\frac{1}{2} + \alpha) h \underline{u_{xx}}\right) t_x \\ &\quad + \left((1+\alpha)\frac{u_{i,j+1} - u_{i,j-1}}{2h} - \alpha \frac{u_{i-1,j+1} - u_{i-1,j-1}}{2h}\right) t_y + O(h^2) \\ &= \frac{1}{h} Tu + h\left(\frac{1}{2} + \alpha\right) t_x u_{xx} + O(h^2)\end{aligned}$$

Coupling Interface Method

CIM2 (Case 1):

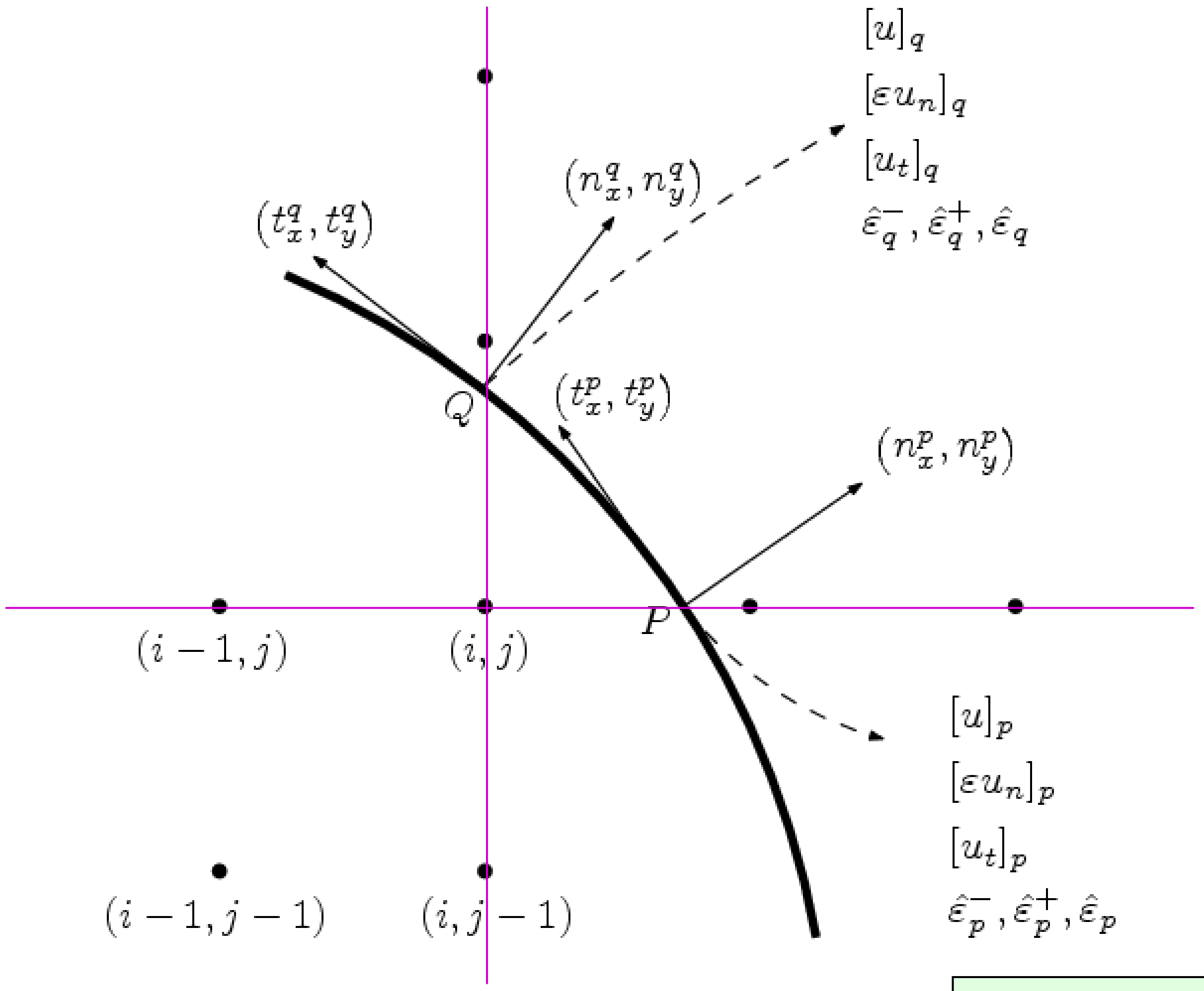
Bounded by 1 and $\varepsilon^+/\varepsilon^-$.

$$\boxed{\left(1-\left(\frac{1}{2}+\alpha\right)a_t t_x\right)} u_{xx} = \frac{1}{h^2}\left(Lu + a_t T u + J\right)$$

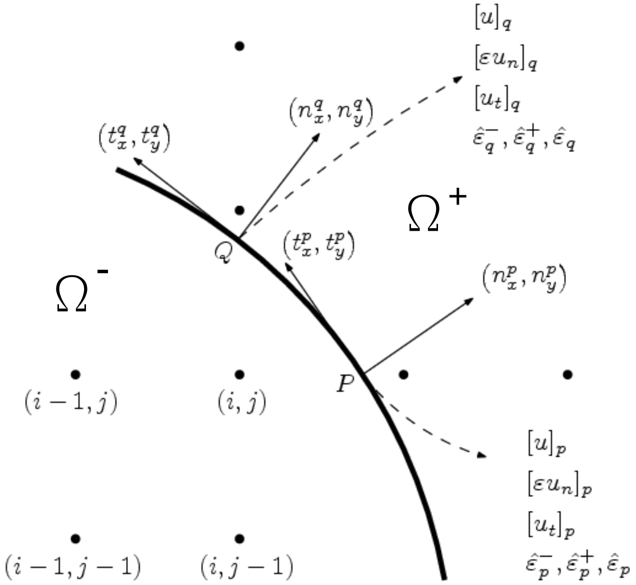
$$a_t = a_\sigma(\rho^+ - \rho^-)t_x,\ \rho^\pm = \hat{\varepsilon}_p^\pm/\hat{\varepsilon}_p,$$

$$J = a_\tau[u]_p + a_\sigma h\left(\frac{[\varepsilon u_n]}{\hat{\varepsilon}} n_x + \rho^+[u_t]t_x\right)$$

CIM2 (Case 2):


$[u]_q$
$[\varepsilon u_n]_q$
$[u_t]_q$
$\hat{\varepsilon}_q^-, \hat{\varepsilon}_q^+, \hat{\varepsilon}_q$
(n_x^q, n_y^q)
(t_x^q, t_y^q)
Q
(t_x^p, t_y^p)
(n_x^p, n_y^p)
P
$(i-1, j)$
(i, j)
$[u]_p$
$[\varepsilon u_n]_p$
$[u_t]_p$
$\hat{\varepsilon}_p^-, \hat{\varepsilon}_p^+, \hat{\varepsilon}_p$
$(i-1, j-1)$
$(i, j-1)$

CIM2 (Case 2):



- Dimension splitting approach

$$\begin{aligned} u_{xx} &= \frac{1}{h^2}\left(L_x u + a_{\tau,p}[u]_p + a_{\sigma,p} h \boxed{\frac{[\varepsilon u_x]_p}{\hat{\varepsilon}_p}}\right) + O(h) \\ u_{yy} &= \frac{1}{h^2}\left(L_y u + a_{\tau,q}[u]_p + a_{\sigma,q} h \boxed{\frac{[\varepsilon u_y]_q}{\hat{\varepsilon}_q}}\right) + O(h) \end{aligned}$$

- Decomposition of jump conditions

$$\begin{aligned} \boxed{[\varepsilon u_x]_p} &= [\varepsilon u_n]_p n_x^p + \left(\hat{\varepsilon}_p^+ [u_t]_p + (\hat{\varepsilon}_p^+ - \hat{\varepsilon}_p^-)\boxed{(u_t^-)_p}\right)(t_x^p) \\ \boxed{[\varepsilon u_y]_q} &= [\varepsilon u_n]_q n_y^q + \left(\hat{\varepsilon}_q^+ [u_t]_q + (\hat{\varepsilon}_q^+ - \hat{\varepsilon}_q^-)\boxed{(u_t^-)_q}\right)(t_y^q) \end{aligned}$$

- One-side interpolation

$$\begin{aligned} \boxed{(u_t^-)_p} &\approx \left(\frac{u_{i,j} - u_{i-1,j}}{h} + (\frac{1}{2} + \alpha_p) h u_{xx}\right) t_x^p + \left((1+\alpha_p)\frac{u_{i,j} - u_{i,j-1}}{h} - \alpha_p \frac{u_{i-1,j} - u_{i-1,j-1}}{h} + \frac{1}{2} h u_{yy}\right) t_y^p \\ \boxed{(u_t^-)_q} &\approx \left(\frac{u_{i,j} - u_{i,j-1}}{h} + (\frac{1}{2} + \alpha_q) h u_{yy}\right) t_y^q + \left((1+\alpha_q)\frac{u_{i,j} - u_{i-1,j}}{h} - \alpha_q \frac{u_{i,j-1} - u_{i-1,j-1}}{h} + \frac{1}{2} h u_{xx}\right) t_x^q \end{aligned}$$

The second order derivatives are coupled by jump conditions

Coupling Interface Method

CIM2 (Case 2): results a coupling matrix

$$\mathbf{M}\begin{bmatrix} u_{xx} \\ u_{yy} \end{bmatrix} = \begin{bmatrix} L_x u + a_{t,p} T_p u + J_p \\ L_y u + a_{t,q} T_q u + J_q \end{bmatrix}$$

$$\begin{aligned}
\mathbf{M} &= \boxed{\begin{bmatrix} 1-(\frac{1}{2}+\alpha_p)a_{t,p}t_x^p & -\frac{1}{2}a_{t,p}t_y^p \\ -\frac{1}{2}a_{t,q}t_x^q & 1-(\frac{1}{2}+\alpha_q)t_y^q \end{bmatrix}} \\
a_{t,p} &= a_{\sigma,p}(\rho_p^+ - \rho_q^-)t_x^p \\
a_{t,q} &= a_{\sigma,q}(\rho_q^+ - \rho_q^-)t_y^q \\
\rho_p^\pm &= \hat{\varepsilon}_p^\pm/\hat{\varepsilon}_p \\
\rho_q^\pm &= \hat{\varepsilon}_q^\pm/\hat{\varepsilon}_q \\
J_p &= a_{\tau,p}[u]_p + a_{\sigma,p}h\left(\frac{[\varepsilon u_n]_p}{\hat{\varepsilon}_p}n_x^p + \rho_p^+[u_t]_p t_x^p\right) \\
J_q &= a_{\tau,q}[u]_q + a_{\sigma,q}h\left(\frac{[\varepsilon u_n]_q}{\hat{\varepsilon}_q}n_y^q + \rho_q^+[u_t]_q t_y^q\right)
\end{aligned}$$

Theorem: det(M) is positive when local curvature is zero or h is small

CIM1: d dimensions

- Dimension splitting approach

$$\begin{aligned}\frac{\partial}{\partial x_k}u(\mathbf{x}+\frac{1}{2}h\mathbf{e}_k) &\approx \frac{1}{h}\left(\bar{\rho}^+_{k_+}(u(\mathbf{x}+h\mathbf{e}_k)-u(\mathbf{x}))-\bar{\rho}^+_{k_+}[u]_{\widehat{\mathbf{x}}_{k_+}}-\beta_{k_+}h\frac{[\varepsilon\nabla u\cdot\mathbf{e}_k]_{\widehat{\mathbf{x}}_{k_+}}}{\bar{\varepsilon}_{k_+}}\right)\\ \frac{\partial}{\partial x_k}u(\mathbf{x}-\frac{1}{2}h\mathbf{e}_k) &\approx \frac{1}{h}\left(\bar{\rho}^+_{k_-}(u(\mathbf{x})-u(\mathbf{x}-h\mathbf{e}_k))+\bar{\rho}^+_{k_-}[u]_{\widehat{\mathbf{x}}_{k_-}}-\beta_{k_-}h\frac{[\varepsilon\nabla u\cdot\mathbf{e}_k]_{\widehat{\mathbf{x}}_{k_-}}}{\bar{\varepsilon}_{k_-}}\right)\end{aligned}$$

- Decomposition of jump conditions

$$[\varepsilon\nabla u\cdot\mathbf{e}_k]_{\widehat{\mathbf{x}}_k} = [\varepsilon\nabla u\cdot\mathbf{n}_k]_{\widehat{\mathbf{x}}_k}(\mathbf{n}_k\cdot\mathbf{e}_k)+[\varepsilon\nabla u\cdot\mathbf{t}_k]_{\widehat{\mathbf{x}}_k}(\mathbf{t}_k\cdot\mathbf{e}_k)$$

$$= \sigma_k(\mathbf{n}_k\cdot\mathbf{e}_k)+\left(\hat{\varepsilon}^+_k[\nabla u\cdot\mathbf{t}_k]_{\widehat{\mathbf{x}}_k}+(\hat{\varepsilon}^+_k-\hat{\varepsilon}^-_k)\nabla u^-(\widehat{\mathbf{x}}_k)\cdot\mathbf{t}_k\right)(\mathbf{t}_k\cdot\mathbf{e}_k)$$

- One-side interpolation
  - $j=k:\ \frac{\partial}{\partial x_k}u^-(\hat{\mathbf{x}}_{k_\pm})\approx\frac{\partial}{\partial x_k}u(\mathbf{x}\pm\frac{1}{2}h\mathbf{e}_k)$
  - $j\neq k:\quad \frac{\partial}{\partial x_j}u^-(\hat{\mathbf{x}}_{k_\pm}) = \begin{cases} D_j^{(s_j)}u(\mathbf{x}) & \text{if } \gamma_{j+\frac{1}{2}}+\gamma_{j-\frac{1}{2}}<2\\ \frac{\partial}{\partial x_j}u^-(\mathbf{x}\pm\frac{1}{2}h\mathbf{e}_j) & \text{if } \gamma_{j+\frac{1}{2}}+\gamma_{j-\frac{1}{2}}=2\end{cases}$

CIM2: d dimensions

- Dimension splitting approach

$$\frac{\partial^2}{\partial x_k^2}u(\mathbf{x}) = \frac{1}{h^2}\left(L_k^{(s_k)}u(\mathbf{x}) + a_{\tau,k}[u]_{\widehat{\mathbf{x}}_k} + s_k a_{\sigma,k} h \frac{[\varepsilon\nabla u\cdot\mathbf{e}_k]_{\widehat{\mathbf{x}}_k}}{\hat{\varepsilon}_k}\right) + O(h)$$

- Decomposition of jump conditions

$$[\varepsilon\nabla u\cdot\mathbf{e}_k]_{\widehat{\mathbf{x}}_k} = [\varepsilon\nabla u\cdot\mathbf{n}_k]_{\widehat{\mathbf{x}}_k}(\mathbf{n}_k\cdot\mathbf{e}_k) + [\varepsilon\nabla u\cdot\mathbf{t}_k]_{\widehat{\mathbf{x}}_k}(\mathbf{t}_k\cdot\mathbf{e}_k)$$

$$= \sigma_k(\mathbf{n}_k\cdot\mathbf{e}_k) + \left(\hat{\varepsilon}_k^+[\nabla u\cdot\mathbf{t}_k]_{\widehat{\mathbf{x}}_k} + (\hat{\varepsilon}_k^+ - \hat{\varepsilon}_k^-)\nabla u^-(\widehat{\mathbf{x}}_k)\cdot\mathbf{t}_k\right)(\mathbf{t}_k\cdot\mathbf{e}_k)$$

- One-side interpolation

$$\begin{aligned}&\nabla u^-(\widehat{\mathbf{x}}_k)\cdot\mathbf{t}_k\\ = \quad &\frac{1}{h}T_k u(\mathbf{x}) + h\left(s_k(\frac{1}{2}+\alpha_k)(\mathbf{t}_k\cdot\mathbf{e}_k)\frac{\partial^2}{\partial x_k^2}u(\mathbf{x}) + \frac{1}{2}\sum_{j=1,j\neq k}^{d} s_j(\mathbf{t}_k\cdot\mathbf{e}_j)\frac{\partial^2}{\partial x_j^2}u(\mathbf{x})\right)\end{aligned}$$

CIM2: d dimensions, coupling matrix

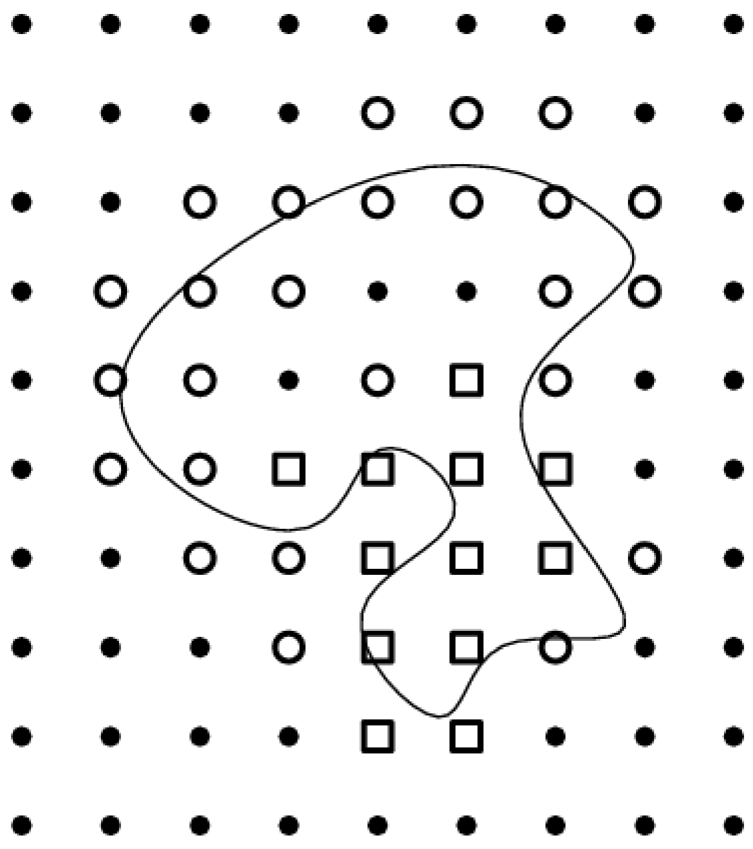
$$\mathbf{M}\left(\frac{\partial^2}{\partial x_k^2}u(\mathbf{x})\right)_{k=1}^{d} = \frac{1}{h^2}(Lu(\mathbf{x}) + Tu(\mathbf{x}) + J),$$

$$m_{k,j} = \begin{cases} 1 - |s_k|(\frac{1}{2} + \alpha_k)a_{t,k}(\mathbf{t}_k \cdot \mathbf{e}_k) & j = k \\ -\frac{1}{2}s_j s_k a_{t,k}(\mathbf{t}_k \cdot \mathbf{e}_j) & j \neq k, \end{cases}$$

$$\begin{aligned} L &= (L_1, \cdots, L_d)^T, \\ T &= (s_1 a_{t,1} T_1, \cdots, s_d a_{t,d} T_d)^T \\ J &= (J_1, \cdots, J_d)^T. \end{aligned}$$

Complex interface problems Classification of grid

- Interior points (bullet) (contral finite difference)
    - Nearest neighbors are in the same side
- On-front points (circle and box)
    - Normal (circle) (CIM2).
    - Exceptional (box) (CIM1).



Classification of grids for complex interface (number of grids)

- Interior grids: $O(h^{-d})$
- Normal on-fronts (CIM2): $O(h^{1-d})$
- Exceptional (CIM1): $O(1)$

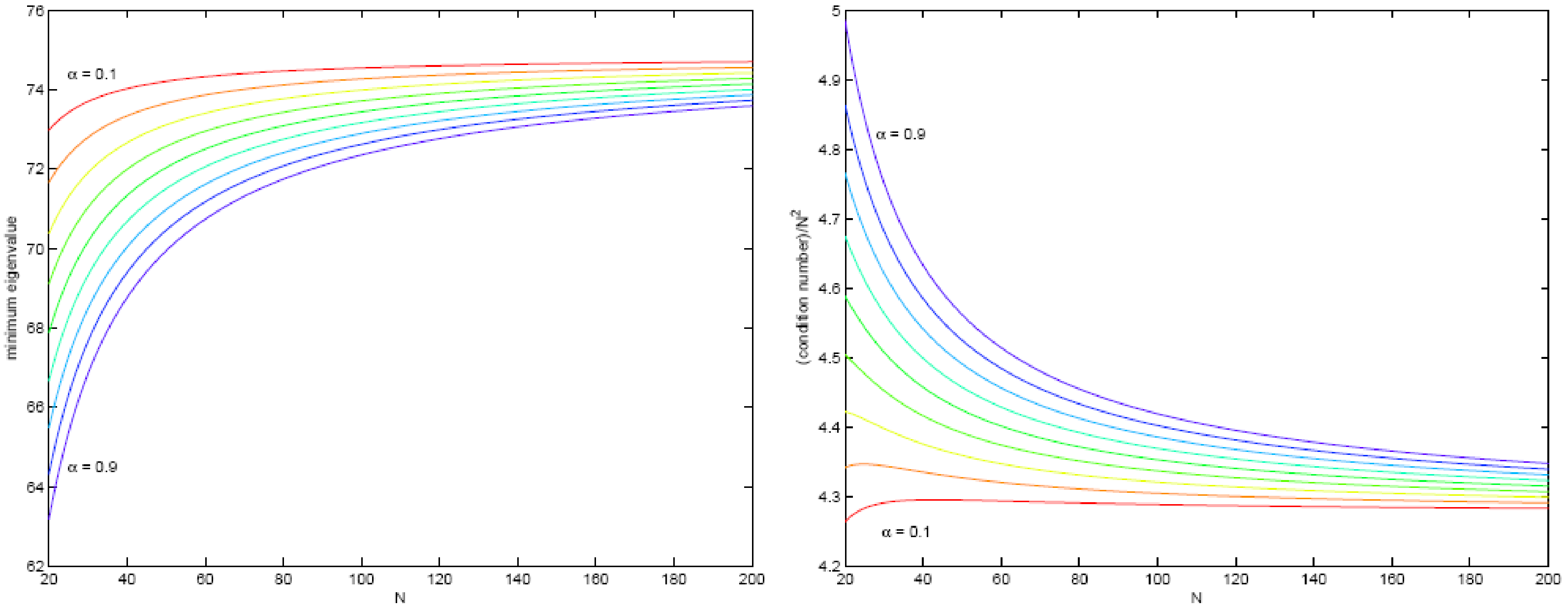
- The resulting scheme is still 2nd order

Numerical Validation

- Stability of CIM2 in 1d
- Orientation error of CIM2 in 2d
- Convergence tests of CIM1
- Comparison results (CIM2)
- Complex interfaces results (Hybrid CIM)

Stability Issue of CIM2 in 1-d

Let $A(\alpha, N)$ be the resulting matrix.

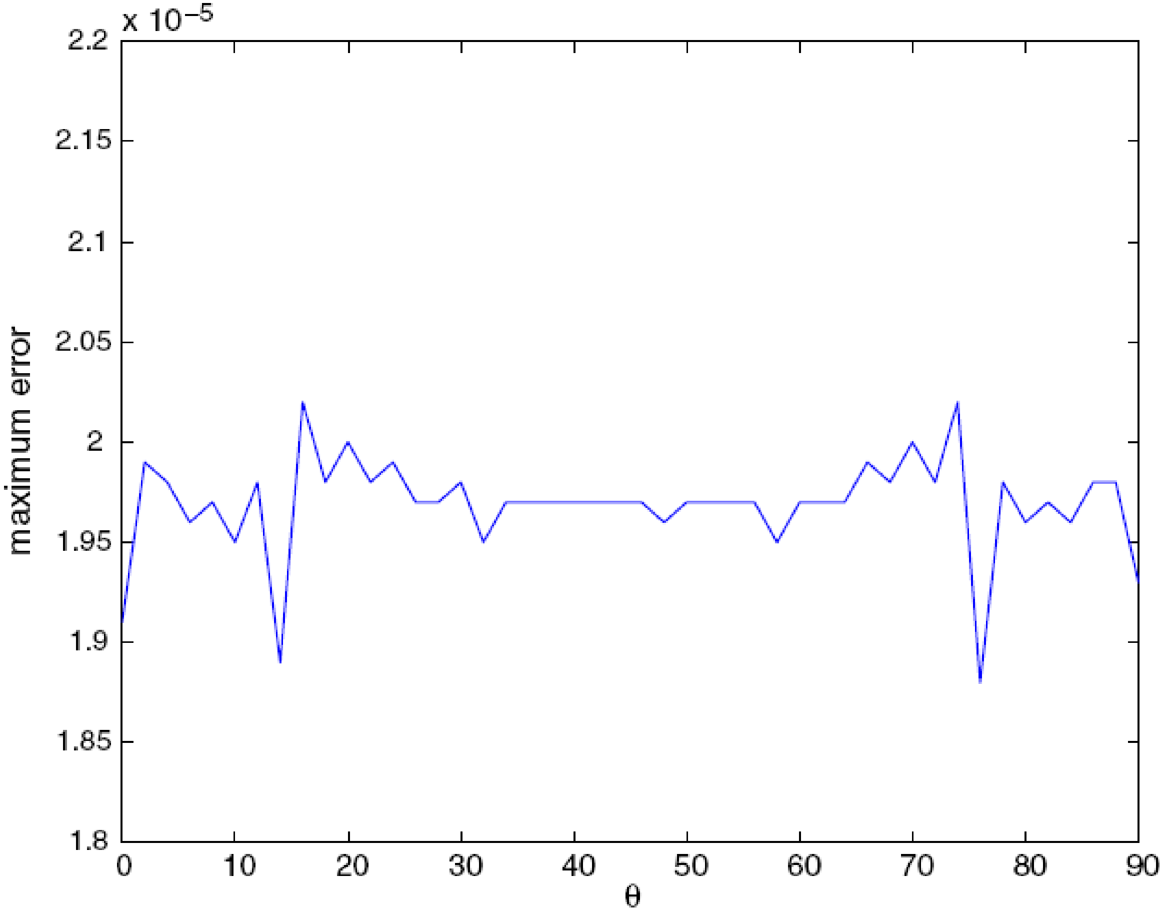


(a) Minimal eigenvalue $\lambda_{min}(\alpha, N)$

(b) Scaled condition number $condA(\alpha, N)/N^2$

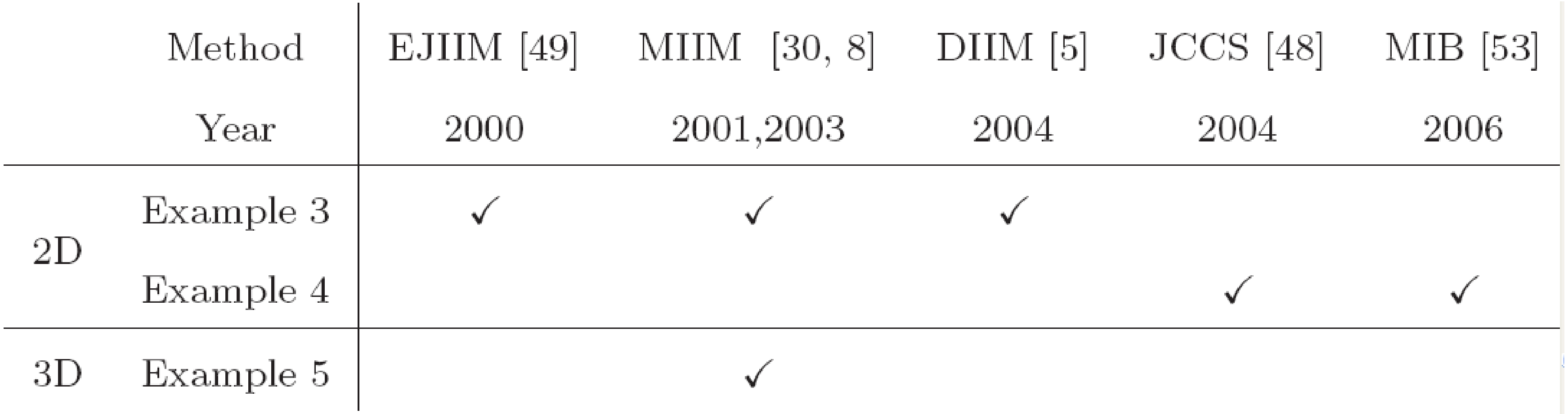
Insensitive to the location of the interface in a cell.

Orientation error from CIM2 is small



Insensitive to the orientation of the interface.

Comparison Table (for CIM2)

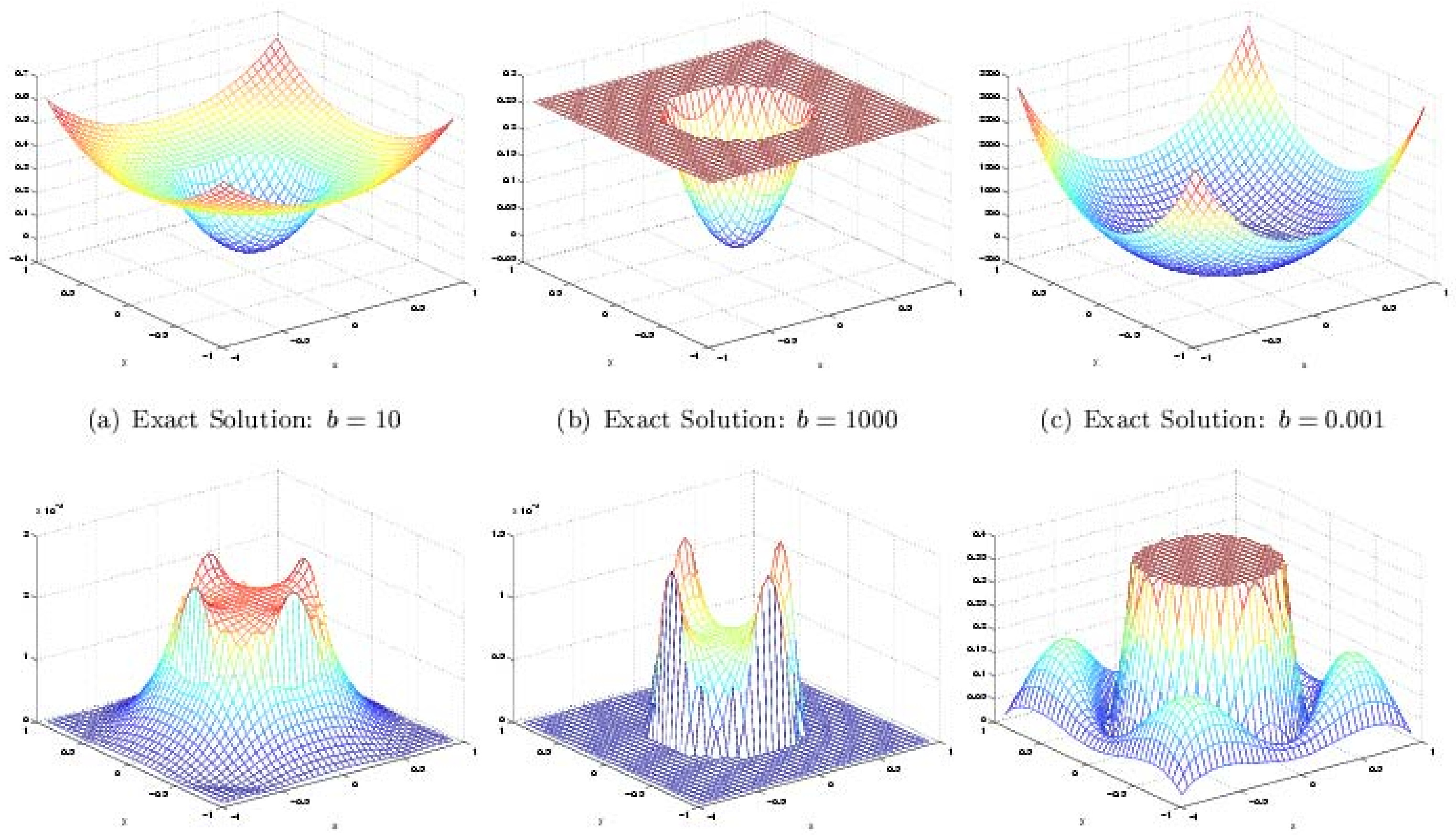


	Method	EJIIM [49]	MIIM [30, 8]	DIIM [5]	JCCS [48]	MIB [53]
	Year	2000	2001,2003	2004	2004	2006
2D	Example 3	✓	✓	✓		
	Example 4				✓	✓
3D	Example 5		✓			

Example 3 (for CIM2)

$$
\begin{aligned}
\phi(x,y) &= r-0.5,\ \ \Omega^- = \{(x,y)|\phi(x,y)<0\},\ \ \Omega^+ = \{(x,y)|\phi(x,y)>0\},\\
\varepsilon(x,y) &= \begin{cases} 1+r^2 & (x,y)\in\Omega^- \\ b & (x,y)\in\Omega^+ \end{cases}\\
u_e(x,y) &= \begin{cases} r^2 & (x,y)\in\Omega^- \\ (r^4/2+r^2+0.1\log(2r))/b-(0.5^4/2+0.5^2)/b+0.5^2 & (x,y)\in\Omega^+ \end{cases}\\
f(x,y) &= -8r^2-4,
\end{aligned}
$$

Example 3, figures



(a) Exact Solution: $b = 10$

(b) Exact Solution: $b = 1000$

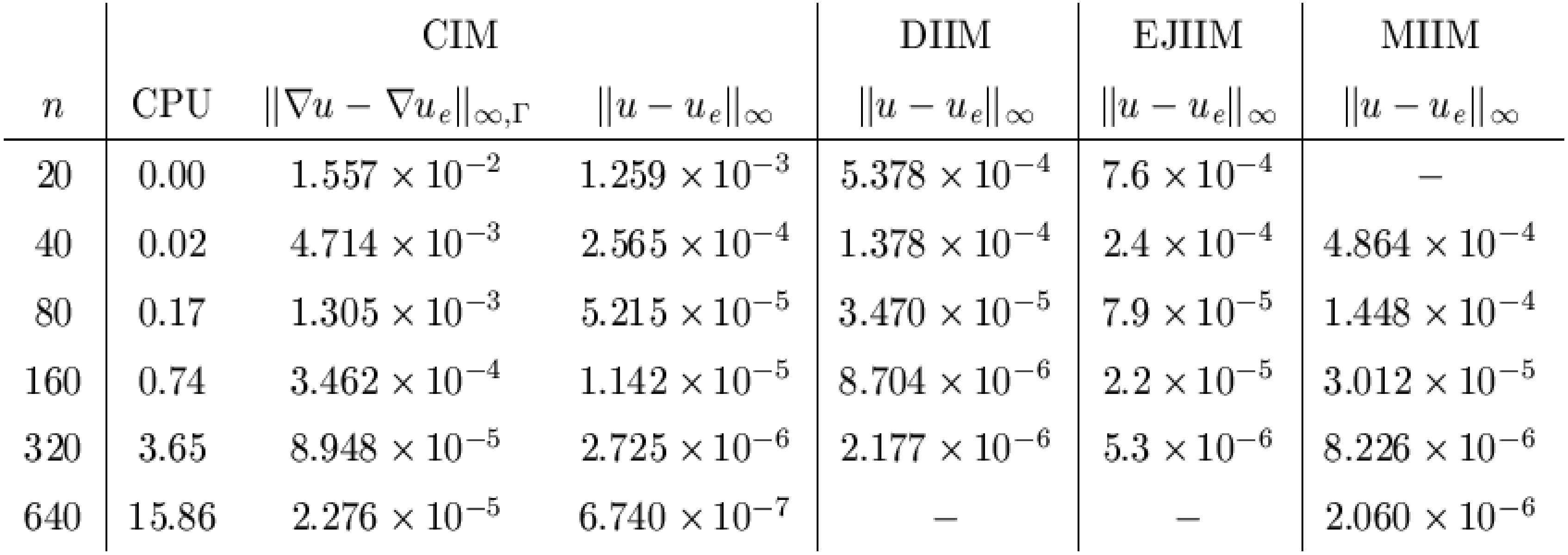
(c) Exact Solution: $b = 0.001$

(d) $|u - u_e|$: $b = 10$

(e) $|u - u_e|$: $b = 1000$

(f) $|u - u_e|$: $b = 0.001$

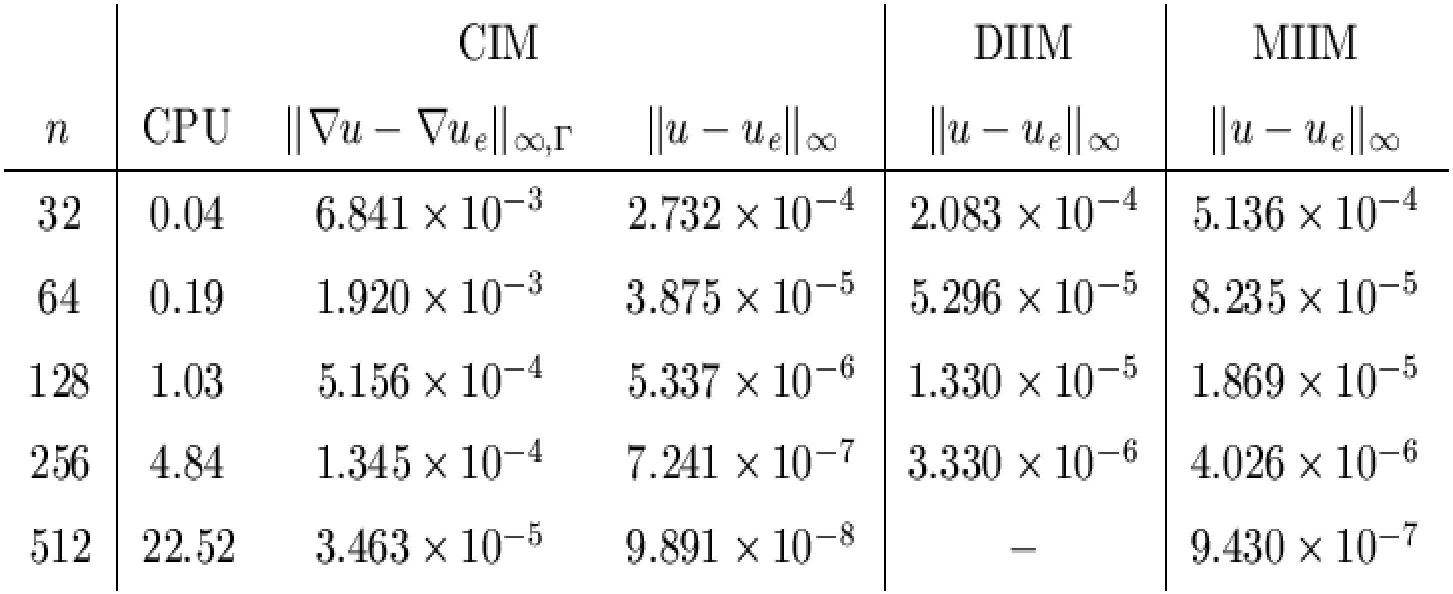
Example 3, Comparison table 1



	CIM			DIIM	EJIIM	MIIM
n	CPU	$\|\|\nabla u - \nabla u_e\|\|_{\infty,\Gamma}$	$\|\|u - u_e\|\|_\infty$	$\|\|u - u_e\|\|_\infty$	$\|\|u - u_e\|\|_\infty$	$\|\|u - u_e\|\|_\infty$
20	0.00	1.557×10^{-2}	1.259×10^{-3}	5.378×10^{-4}	7.6×10^{-4}	–
40	0.02	4.714×10^{-3}	2.565×10^{-4}	1.378×10^{-4}	2.4×10^{-4}	4.864×10^{-4}
80	0.17	1.305×10^{-3}	5.215×10^{-5}	3.470×10^{-5}	7.9×10^{-5}	1.448×10^{-4}
160	0.74	3.462×10^{-4}	1.142×10^{-5}	8.704×10^{-6}	2.2×10^{-5}	3.012×10^{-5}
320	3.65	8.948×10^{-5}	2.725×10^{-6}	2.177×10^{-6}	5.3×10^{-6}	8.226×10^{-6}
640	15.86	2.276×10^{-5}	6.740×10^{-7}	–	–	2.060×10^{-6}

Table 2: Example 1: $b = 10$

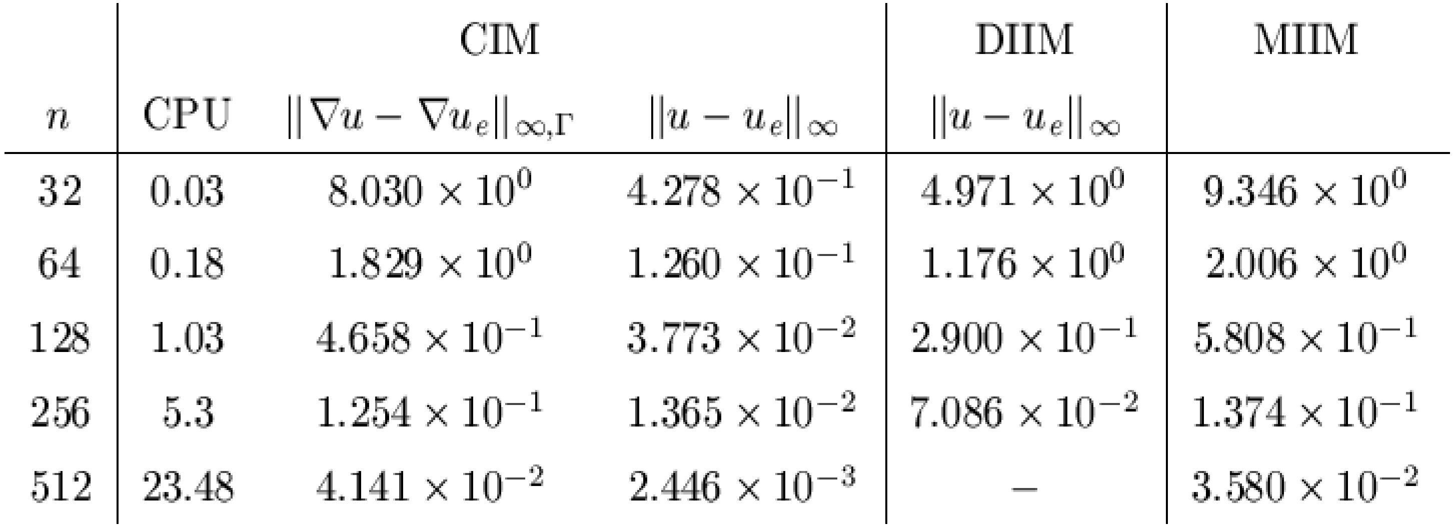
Example 3, Comparison table 2



	CIM			DIIM	MIIM
n	CPU	$\|\|\nabla u - \nabla u_e\|\|_{\infty,\Gamma}$	$\|\|u - u_e\|\|_\infty$	$\|\|u - u_e\|\|_\infty$	$\|\|u - u_e\|\|_\infty$
32	0.04	6.841×10^{-3}	2.732×10^{-4}	2.083×10^{-4}	5.136×10^{-4}
64	0.19	1.920×10^{-3}	3.875×10^{-5}	5.296×10^{-5}	8.235×10^{-5}
128	1.03	5.156×10^{-4}	5.337×10^{-6}	1.330×10^{-5}	1.869×10^{-5}
256	4.84	1.345×10^{-4}	7.241×10^{-7}	3.330×10^{-6}	4.026×10^{-6}
512	22.52	3.463×10^{-5}	9.891×10^{-8}	–	9.430×10^{-7}

Table 3: Example 1: $b = 1000$

Example 3 for CIM2



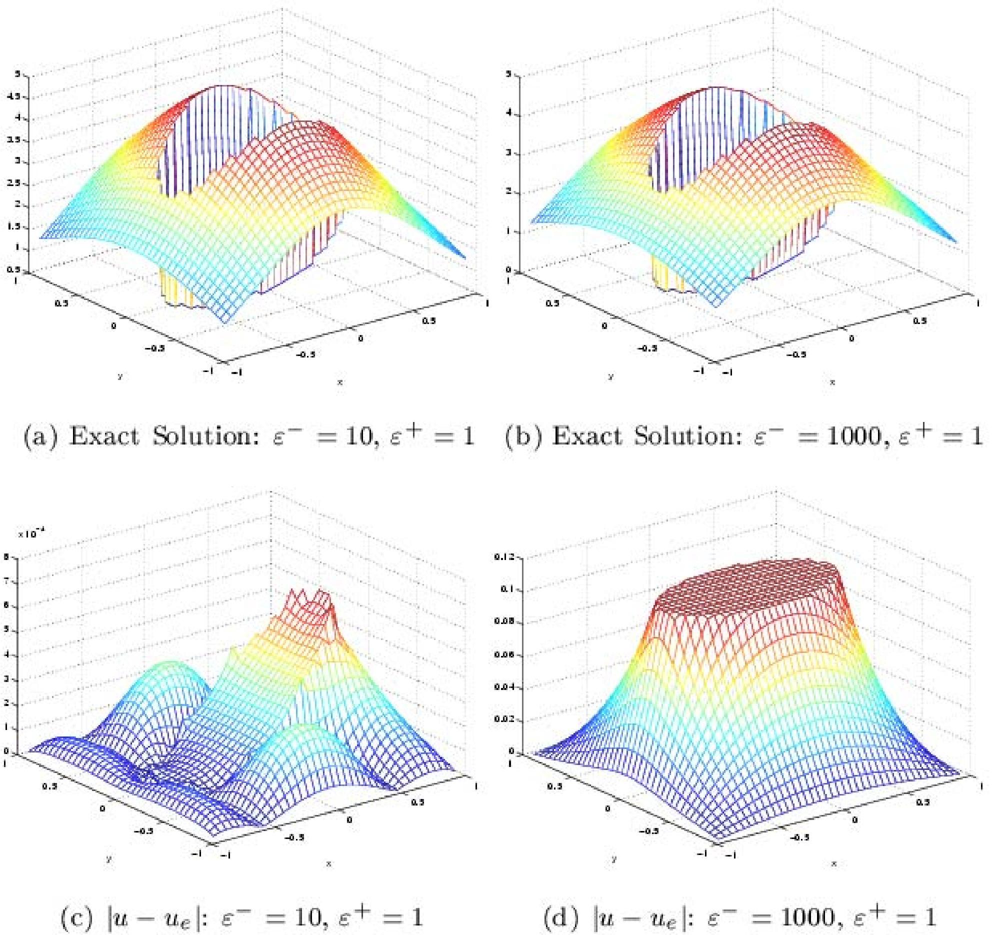
	CIM			DIIM	MIIM
n	CPU	$\|\|\nabla u - \nabla u_e\|\|_{\infty,\Gamma}$	$\|\|u - u_e\|\|_\infty$	$\|\|u - u_e\|\|_\infty$	
32	0.03	8.030×10^0	4.278×10^{-1}	4.971×10^0	9.346×10^0
64	0.18	1.829×10^0	1.260×10^{-1}	1.176×10^0	2.006×10^0
128	1.03	4.658×10^{-1}	3.773×10^{-2}	2.900×10^{-1}	5.808×10^{-1}
256	5.3	1.254×10^{-1}	1.365×10^{-2}	7.086×10^{-2}	1.374×10^{-1}
512	23.48	4.141×10^{-2}	2.446×10^{-3}	–	3.580×10^{-2}

Table 4: Example 1: $b = 0.001$

Example 4 (for CIM2)

$$
\begin{aligned}
\phi(x,y) &= \left(\frac{x^2}{18/27}\right)^2 + \left(\frac{y}{10/27}\right)^2 - 1,\ \ \Omega^- = \{(x,y)|\phi(x,y) < 0\},\ \ \Omega^+ = \{(x,y)|\phi(x,y) > 0\} \\
\varepsilon(x,y) &= \begin{cases} \varepsilon^- & (x,y) \in \Omega^- \\ \varepsilon^+ & (x,y) \in \Omega^+ \end{cases} \\
u_e(x,y) &= \begin{cases} e^x \cos y & (x,y) \in \Omega^- \\ 5e^{-x^2-y^2/2} & (x,y) \in \Omega^+ \end{cases} \\
f(\mathbf{x}) &= \begin{cases} 0 & (x,y) \in \Omega^- \\ -5e^{-x^2-y^2/2}(-3+4x^2+y^2); & (x,y) \in \Omega^+ \end{cases}
\end{aligned}
$$

Example 4, figures



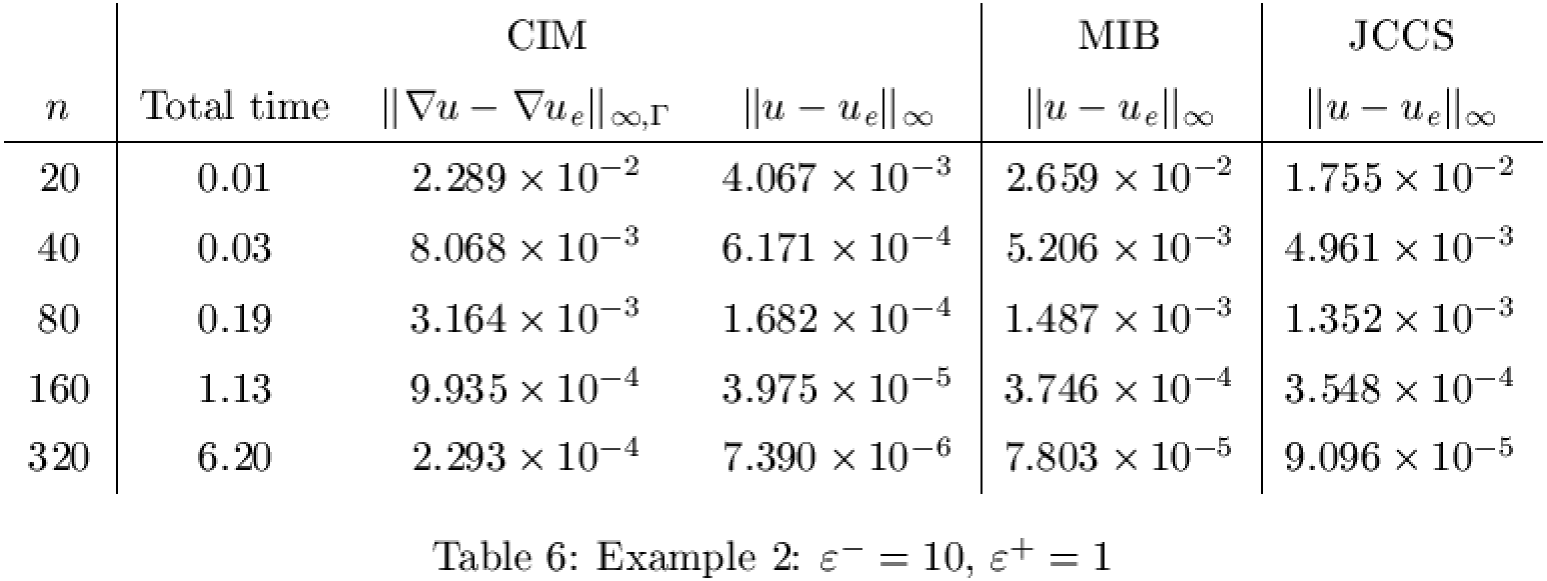
(a) Exact Solution: $\varepsilon^- = 10$, $\varepsilon^+ = 1$

(b) Exact Solution: $\varepsilon^- = 1000$, $\varepsilon^+ = 1$

(c) $|u - u_e|$: $\varepsilon^- = 10$, $\varepsilon^+ = 1$

(d) $|u - u_e|$: $\varepsilon^- = 1000$, $\varepsilon^+ = 1$

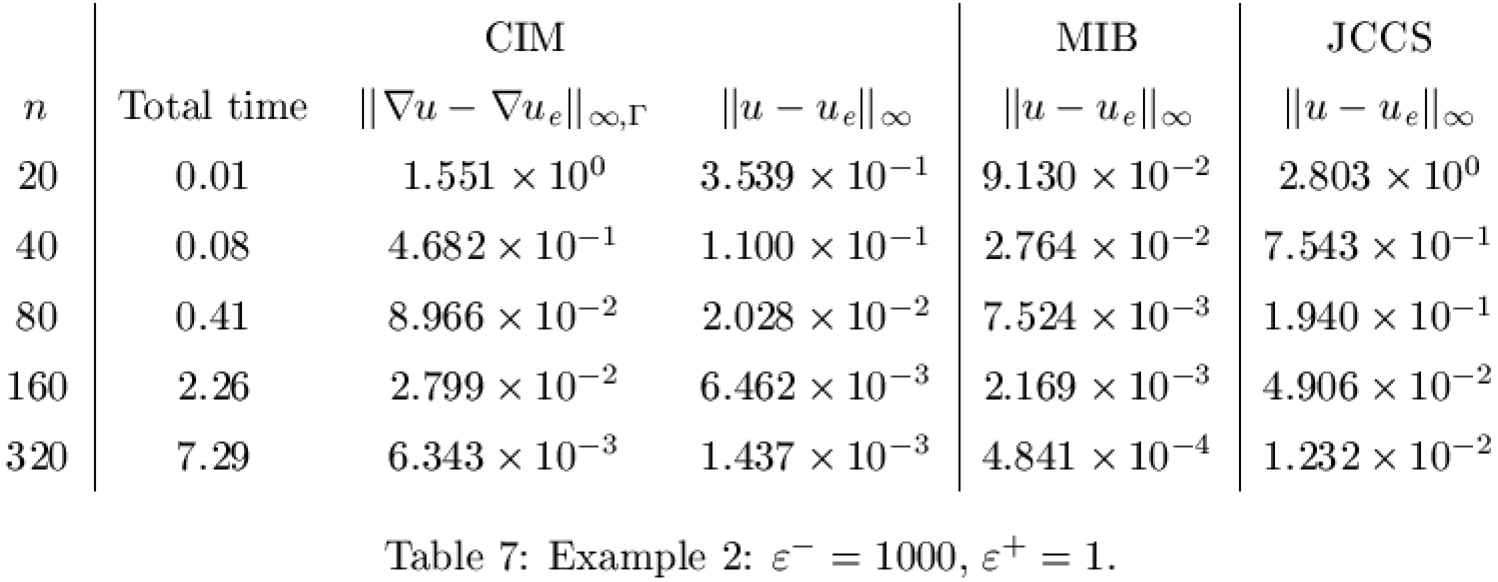
Example 4 (for CIM2)



	CIM			MIB	JCCS
n	Total time	$\|\|\nabla u - \nabla u_e\|\|_{\infty,\Gamma}$	$\|\|u - u_e\|\|_\infty$	$\|\|u - u_e\|\|_\infty$	$\|\|u - u_e\|\|_\infty$
20	0.01	2.289×10^{-2}	4.067×10^{-3}	2.659×10^{-2}	1.755×10^{-2}
40	0.03	8.068×10^{-3}	6.171×10^{-4}	5.206×10^{-3}	4.961×10^{-3}
80	0.19	3.164×10^{-3}	1.682×10^{-4}	1.487×10^{-3}	1.352×10^{-3}
160	1.13	9.935×10^{-4}	3.975×10^{-5}	3.746×10^{-4}	3.548×10^{-4}
320	6.20	2.293×10^{-4}	7.390×10^{-6}	7.803×10^{-5}	9.096×10^{-5}

Table 6: Example 2: $\varepsilon^- = 10$, $\varepsilon^+ = 1$

Example 4, Comparison table 2



		CIM		MIB	JCCS
n	Total time	$\|\|\nabla u - \nabla u_e\|\|_{\infty,\Gamma}$	$\|\|u - u_e\|\|_\infty$	$\|\|u - u_e\|\|_\infty$	$\|\|u - u_e\|\|_\infty$
20	0.01	1.551×10^{0}	3.539×10^{-1}	9.130×10^{-2}	2.803×10^{0}
40	0.08	4.682×10^{-1}	1.100×10^{-1}	2.764×10^{-2}	7.543×10^{-1}
80	0.41	8.966×10^{-2}	2.028×10^{-2}	7.524×10^{-3}	1.940×10^{-1}
160	2.26	2.799×10^{-2}	6.462×10^{-3}	2.169×10^{-3}	4.906×10^{-2}
320	7.29	6.343×10^{-3}	1.437×10^{-3}	4.841×10^{-4}	1.232×10^{-2}

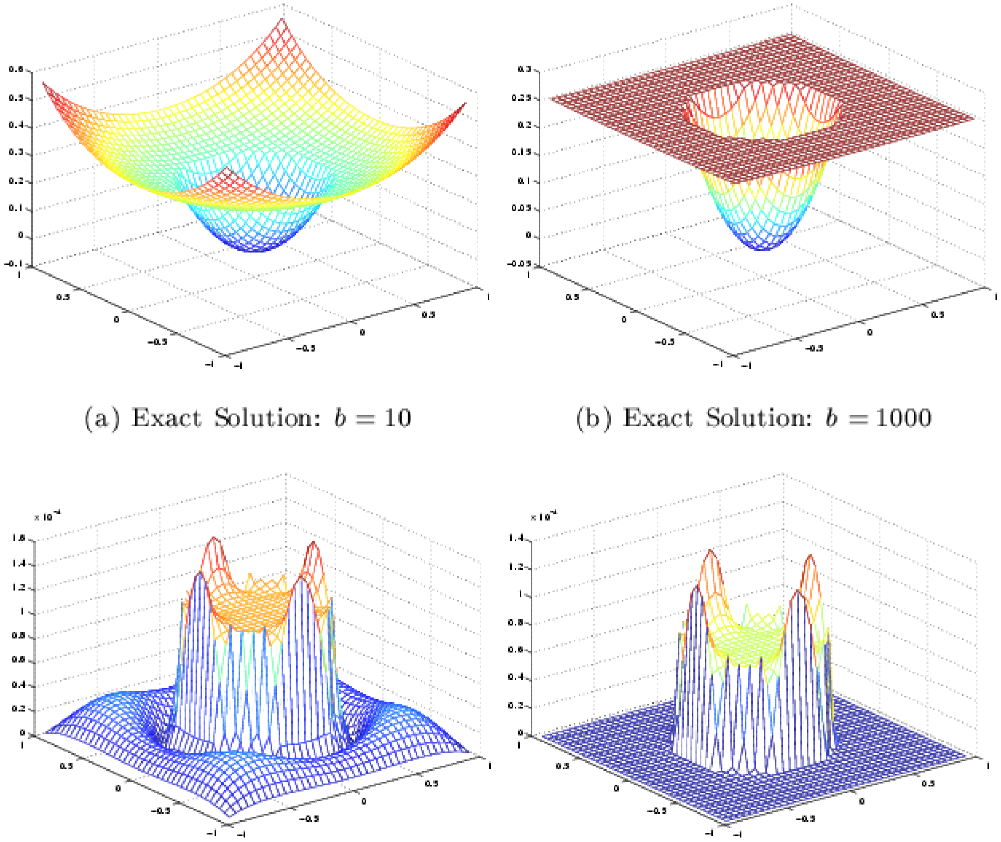
Table 7: Example 2: $\varepsilon^- = 1000$, $\varepsilon^+ = 1$.

Example 5 (for CIM2)

- 3 dimensions

$$
\begin{aligned}
\phi(x,y,z) &= r-0.5,\ \ \Omega^- = \{(x,y,z)|\phi(x,y,z)<0\},\ \ \Omega^+ = \{(x,y,z)|\phi(x,y,z)>0\} \\
\varepsilon(x,y,z) &= \begin{cases} 1+r^2 & (x,y,z)\in\Omega^- \\ b & (x,y,z)\in\Omega^+ \end{cases} \\
u_e(x,y,z) &= \begin{cases} r^2 & (x,y,z)\in\Omega^- \\ (r^4/2+r^2)/b-(0.5^4/2+0.5^2)/b+0.5^2 & (x,y,z)\in\Omega^+ \end{cases} \\
f(x,y,z) &= -(10r^2+6),
\end{aligned}
$$

Example 5, figures



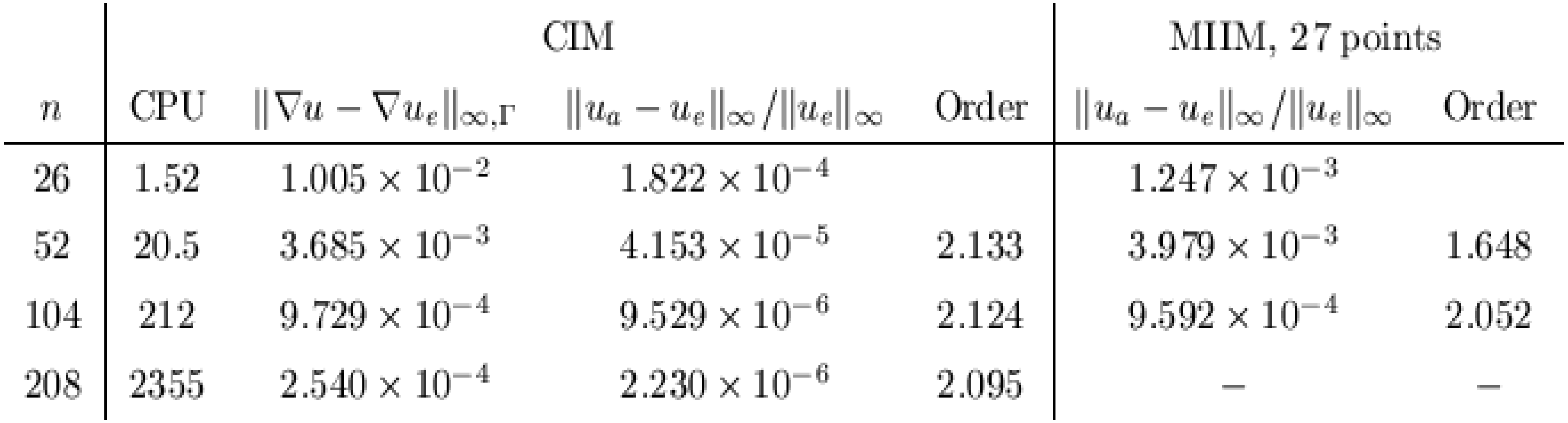
(a) Exact Solution: $b = 10$

(b) Exact Solution: $b = 1000$

(c) $|u - u_e|$: $b = 10$

(d) $|u - u_e|$: $b = 1000$

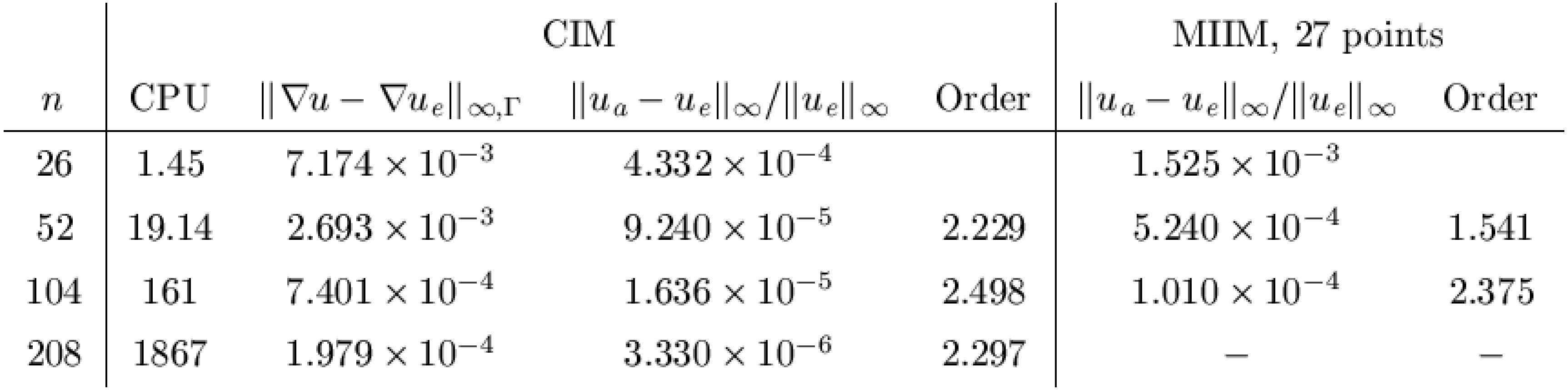
Example 5 (for CIM2)



	CIM				MIIM, 27 points	
n	CPU	$\|\nabla u - \nabla u_e\|_{\infty,\Gamma}$	$\|u_a - u_e\|_\infty / \|u_e\|_\infty$	Order	$\|u_a - u_e\|_\infty / \|u_e\|_\infty$	Order
26	1.52	1.005×10^{-2}	1.822×10^{-4}		1.247×10^{-3}	
52	20.5	3.685×10^{-3}	4.153×10^{-5}	2.133	3.979×10^{-3}	1.648
104	212	9.729×10^{-4}	9.529×10^{-6}	2.124	9.592×10^{-4}	2.052
208	2355	2.540×10^{-4}	2.230×10^{-6}	2.095	–	–

Table 10: Example 4: $b = 1$

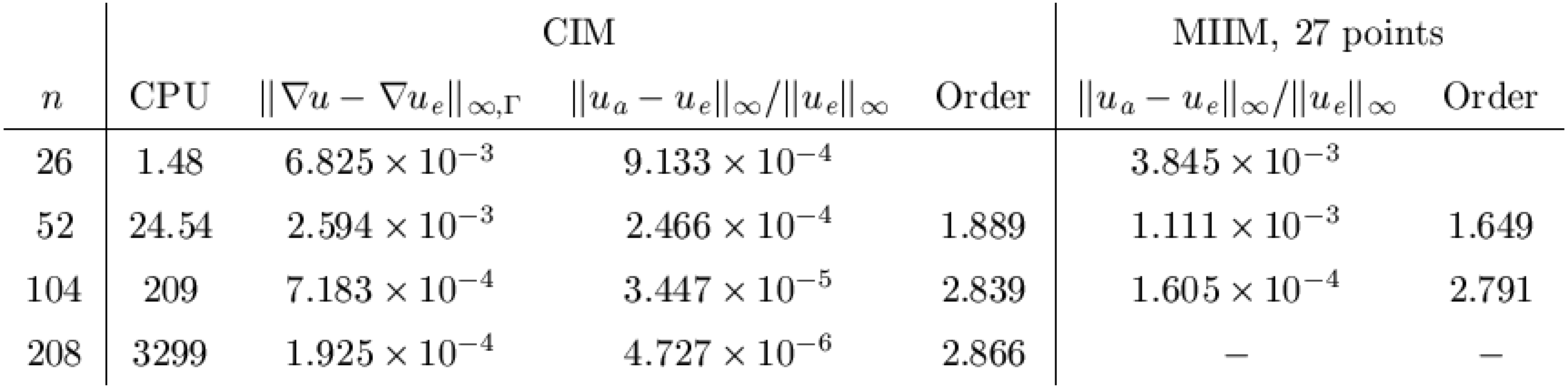
Example 5 (CIM2)



	CIM				MIIM, 27 points	
n	CPU	$\|\|\nabla u - \nabla u_e\|\|_{\infty,\Gamma}$	$\|\|u_a - u_e\|\|_\infty / \|\|u_e\|\|_\infty$	Order	$\|\|u_a - u_e\|\|_\infty / \|\|u_e\|\|_\infty$	Order
26	1.45	7.174×10^{-3}	4.332×10^{-4}		1.525×10^{-3}	
52	19.14	2.693×10^{-3}	9.240×10^{-5}	2.229	5.240×10^{-4}	1.541
104	161	7.401×10^{-4}	1.636×10^{-5}	2.498	1.010×10^{-4}	2.375
208	1867	1.979×10^{-4}	3.330×10^{-6}	2.297	–	–

Table 11: Example 4: $b = 10$

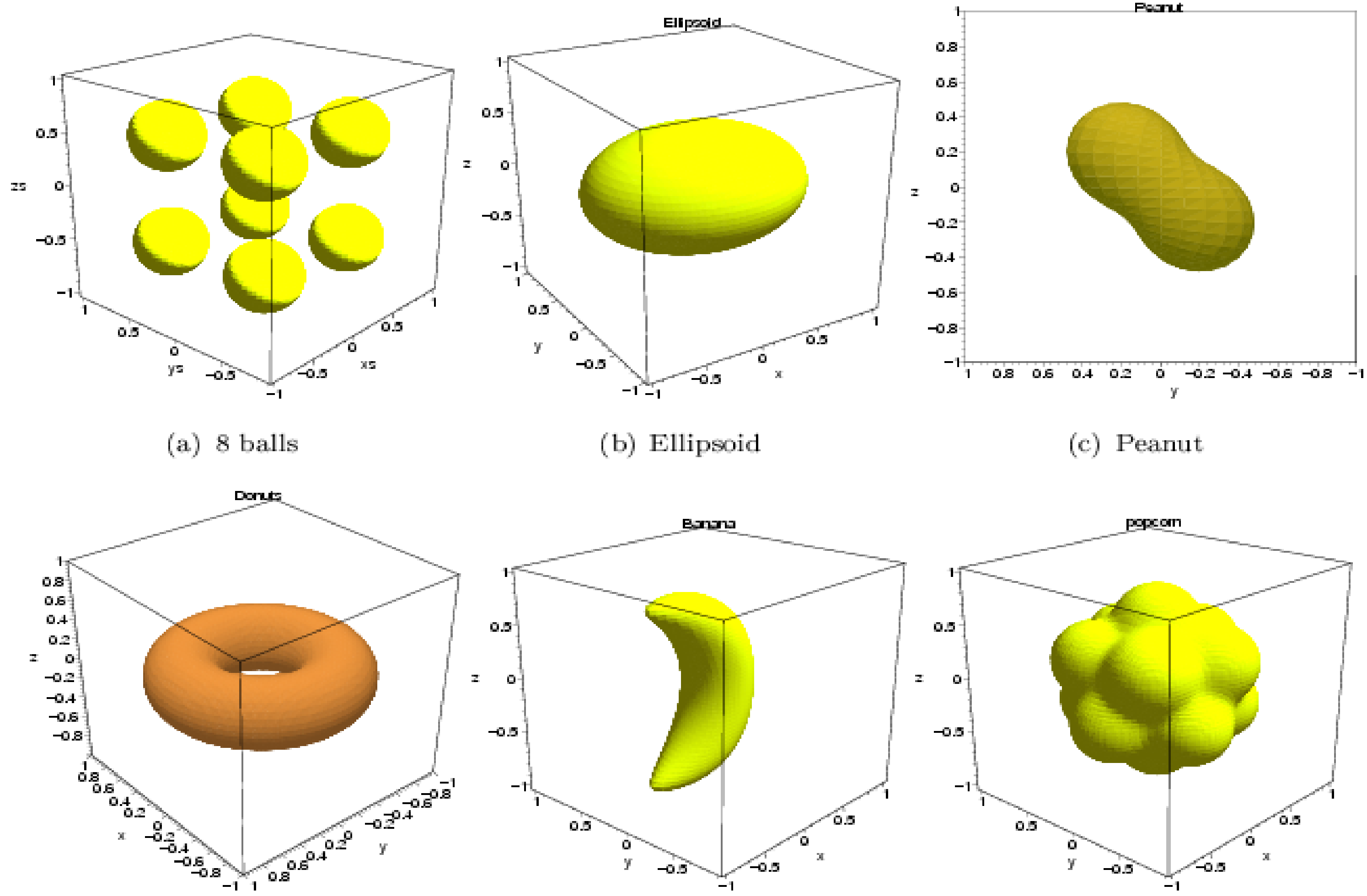
Example 5 (CIM2)



	CIM				MIIM, 27 points	
n	CPU	$\|\|\nabla u - \nabla u_e\|\|_{\infty,\Gamma}$	$\|\|u_a - u_e\|\|_\infty / \|\|u_e\|\|_\infty$	Order	$\|\|u_a - u_e\|\|_\infty / \|\|u_e\|\|_\infty$	Order
26	1.48	6.825×10^{-3}	9.133×10^{-4}		3.845×10^{-3}	
52	24.54	2.594×10^{-3}	2.466×10^{-4}	1.889	1.111×10^{-3}	1.649
104	209	7.183×10^{-4}	3.447×10^{-5}	2.839	1.605×10^{-4}	2.791
208	3299	1.925×10^{-4}	4.727×10^{-6}	2.866	–	–

Table 12: Example 4: $b = 1000$

Convergence tests for CIM1: interfaces



(a) 8 balls

(b) Ellipsoid

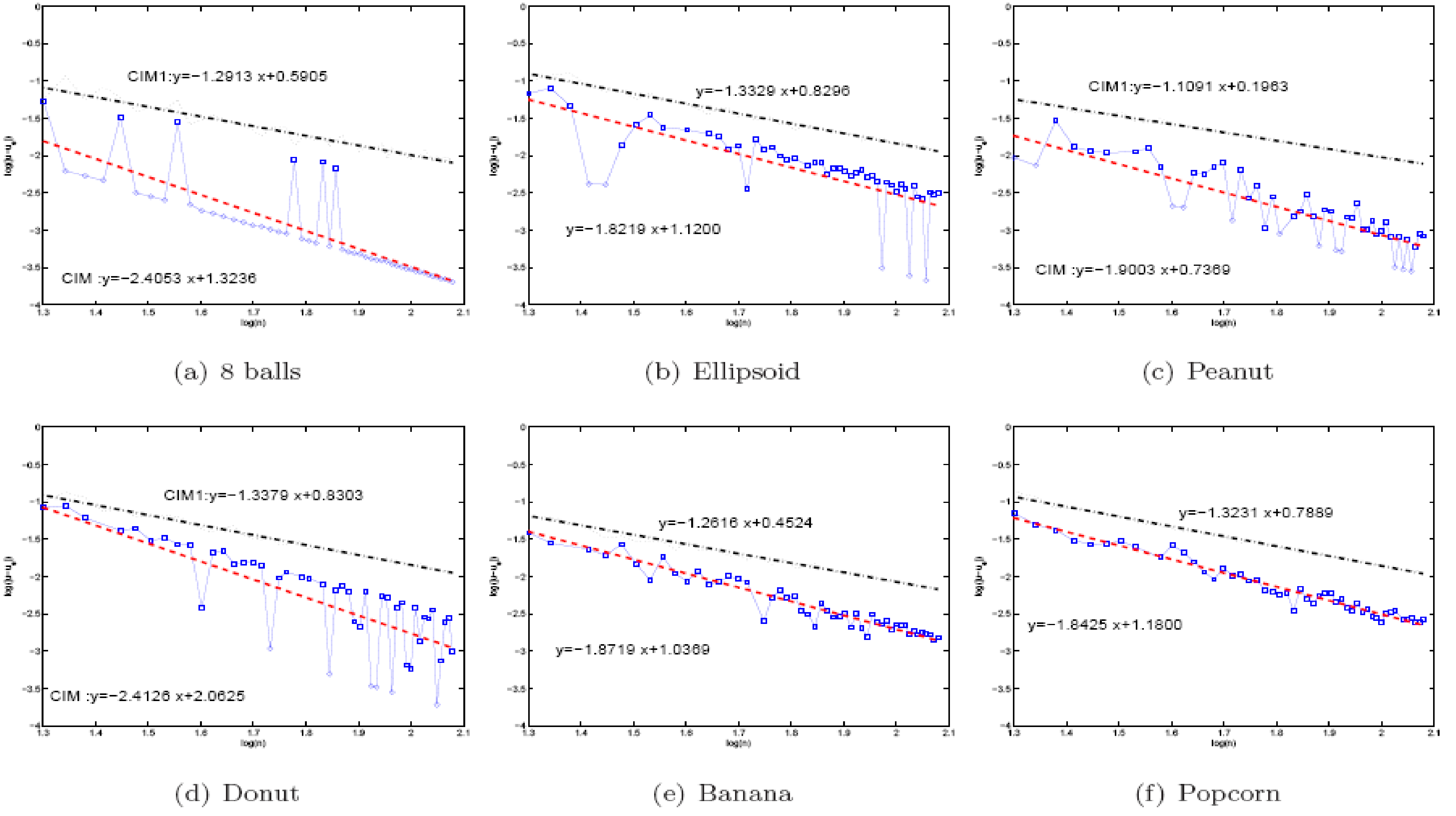
(c) Peanut

(d) Donut

(e) Banana

(f) Popcorn

Convergence of hybrid CIM (order 1.8)



(a) 8 balls

(b) Ellipsoid

(c) Peanut

(d) Donut

(e) Banana

(f) Popcorn

Comparison results

- Second order for u and its gradients in maximum norm for CIM2
- Insensitive to the contrast of epsilon
- Less absolute error despite of using smaller size of stencil
- Linear computational complexity

CIM: Ingredients

- Dimension splitting approach.
- Decomposition of jump conditions
- Coupling: express tangential derivative in terms of principal second derivative to reduce number of interpolation points.
- Solvability of stencil coefficients

CIM:merits

- Accuracy: second-order for u and its gradients in maximum norm with smaller absolute errors than other existing methods.
- Simplicity: smaller size stencil, easy to program.
- Stability: nice stencil coefficients for linear solvers.
- Robustness: capable to handle complex interfaces.
- Speed: linear computational complexity

Poisson-Boltzmann equation

$$-\nabla[\epsilon(r)\nabla\phi(r)] + K(r)\sinh(\phi(r)) = Q(r)$$

$$Q(r) = C\sum_{i=1}^{N_m} q_i\delta(r - r_i), C = \frac{4\pi e_c^2}{k_B T}$$

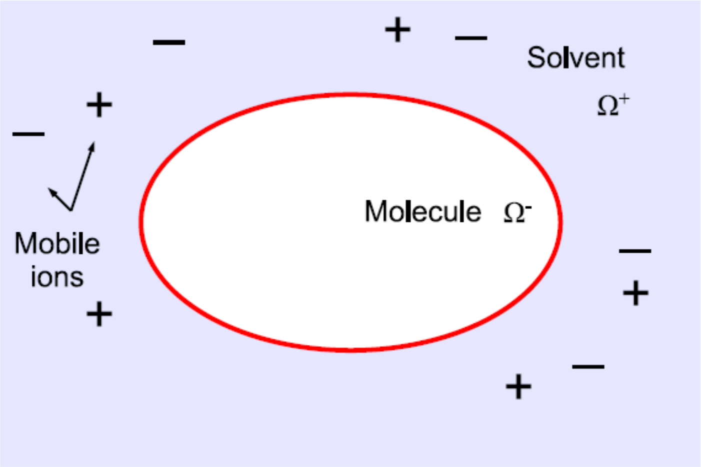
$$\epsilon_1 \approx 1 \sim 2, \epsilon_2 \approx 80,$$

$$5249.0 \leq C \leq 10500.0,$$

$$-1 \leq q_i \leq 1,$$

K =

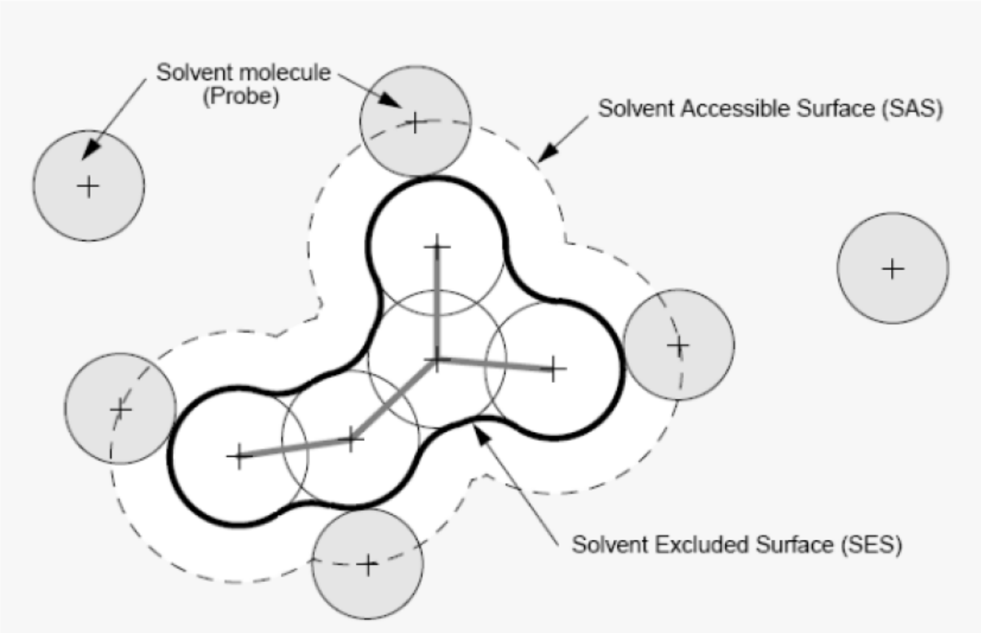
$$\overline{\kappa}^2 = 8.486902807\mathring{A}^{-2} I_s,$$

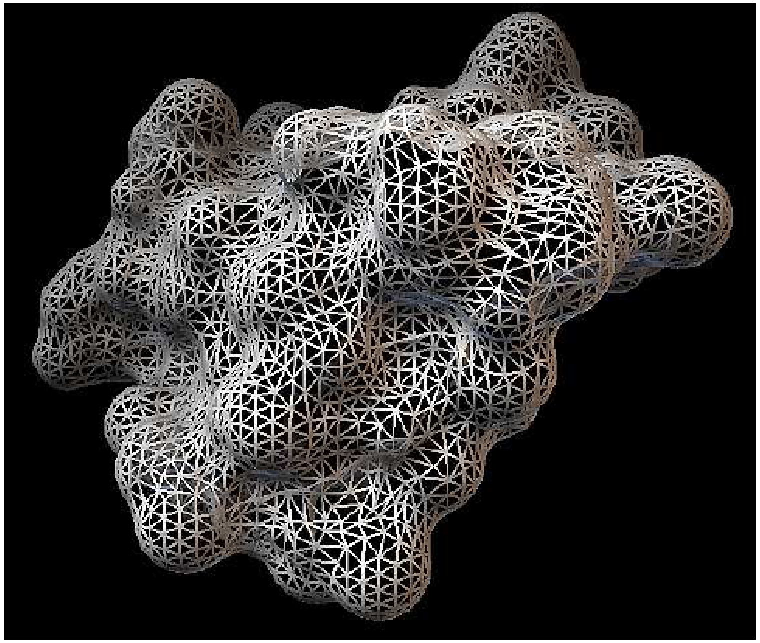


Numerical procedure

- Construction of molecular surface (by MSMS)
- Treatment of singular charges $C \sum q_i(x - x_i)$
- Nonlinear iteration by damped Newton's method for the perturbed equation
- Coupling interface method to solve elliptic interface problem
- Algebraic multigrid for solving linear systems

Construction of molecular surface: MSMS


Solvent molecule (Probe)
Solvent Accessible Surface (SAS)
Solvent Excluded Surface (SES)



The interface calculated by computer software MSMS of molecule 1crn with probe radius 1.4 and triangulation density 3.0.

Treatment of point charge singularities

separate singular part

$$\phi = \overline{\phi} + \tilde{\phi}.$$

where

$$\overline{\phi}(\boldsymbol{x}) = \begin{cases} \phi^*(\boldsymbol{x}) + \phi^0(\boldsymbol{x}) & \boldsymbol{x} \in \Omega_1 \\ 0 & \boldsymbol{x} \in \Omega_2 \cup \Omega_3 \end{cases} .$$

and ϕ^* is the potential in the free space induced by Q, i.e.

$$\phi^*(\boldsymbol{x}) = \begin{cases} C \sum_{i=1}^{m} \frac{1}{\epsilon_1} \frac{z_i}{4\pi} \frac{1}{|\boldsymbol{x} - \boldsymbol{x}_i|}, & \boldsymbol{x} \in R^3 \\ C \sum_{i=1}^{m} -\frac{1}{\epsilon_1} \frac{z_i}{2\pi} \log(|\boldsymbol{x} - \boldsymbol{x}_i|), & \boldsymbol{x} \in R^2 \end{cases} .$$

ϕ^0 is a harmonic function in Ω_1 satisfying

$$\begin{cases} \triangle \phi^0 = 0 & \text{in } \Omega_1 \\ \phi^0 = -\phi^* & \text{on } \Gamma_1. \end{cases}$$

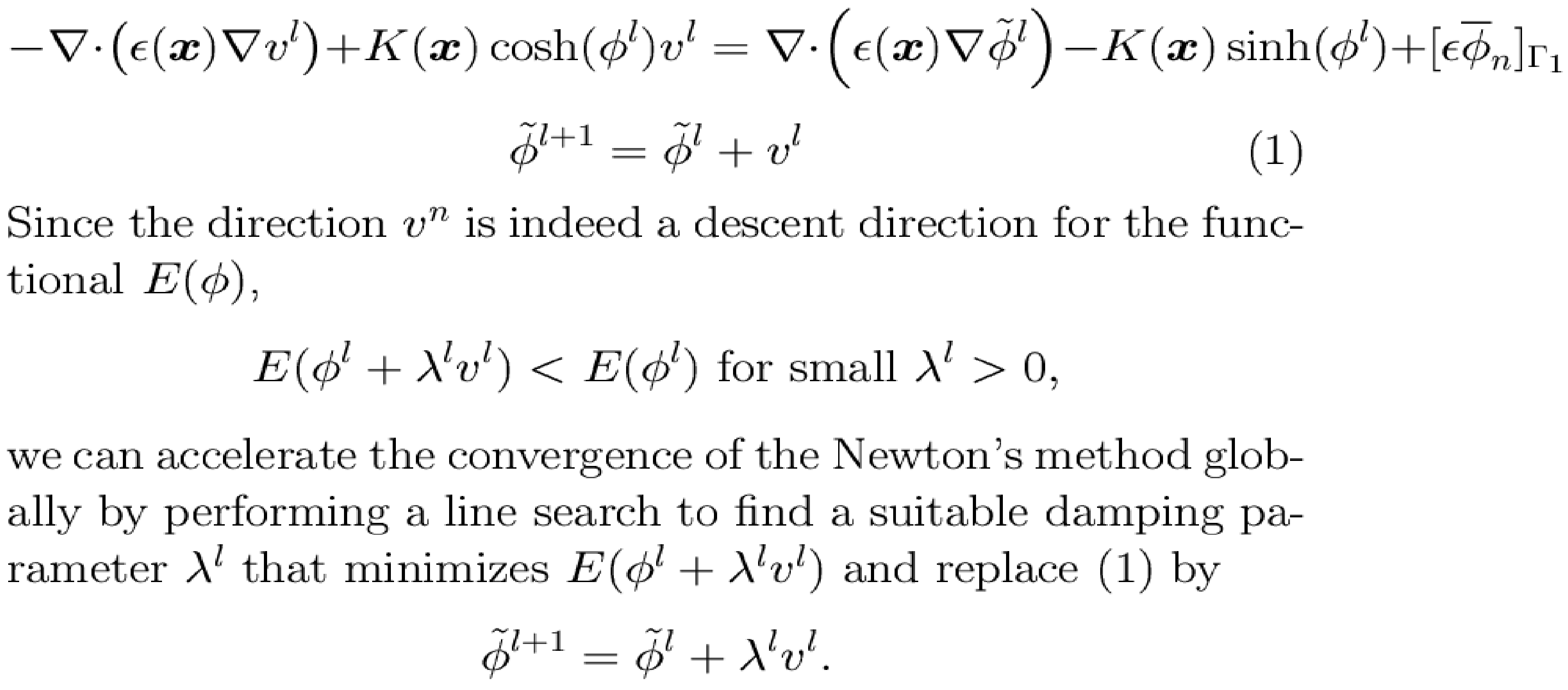
The introduction of ϕ^0 is to force $[\overline{\phi}] = 0$ across Γ_1.

The correction potential satisfies

$$-\nabla \cdot \left(\epsilon(\boldsymbol{x}) \nabla \tilde{\phi}(\boldsymbol{x}) \right) + K(\boldsymbol{x}) \sinh(\tilde{\phi}(\boldsymbol{x})) = [\epsilon \overline{\phi}_n]_{\Gamma_1} \delta_{\Gamma_1}.$$

Thus, the point charge singularity is transfered into surface singularity.

Damped Newton’s Method



$$-\nabla\cdot\left(\epsilon(\boldsymbol{x})\nabla v^l\right)+K(\boldsymbol{x})\cosh(\phi^l)v^l = \nabla\cdot\left(\epsilon(\boldsymbol{x})\nabla\tilde{\phi}^l\right)-K(\boldsymbol{x})\sinh(\phi^l)+[\epsilon\overline{\phi}_n]_{\Gamma_1}$$

$$\tilde{\phi}^{l+1} = \tilde{\phi}^l + v^l \tag{1}$$

Since the direction v^n is indeed a descent direction for the functional $E(\phi)$,

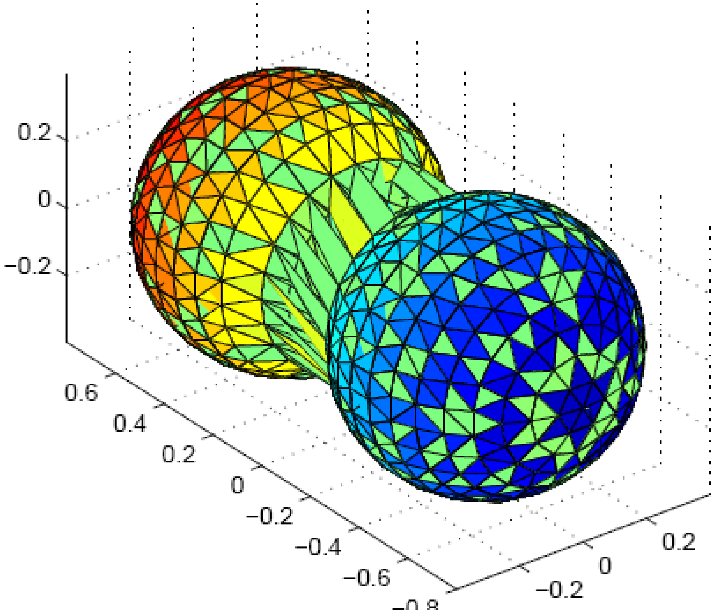
$$E(\phi^l + \lambda^l v^l) < E(\phi^l) \text{ for small } \lambda^l > 0,$$

we can accelerate the convergence of the Newton’s method globally by performing a line search to find a suitable damping parameter λ^l that minimizes $E(\phi^l + \lambda^l v^l)$ and replace (1) by

$$\tilde{\phi}^{l+1} = \tilde{\phi}^l + \lambda^l v^l.$$

Numerical Validation—Artificial molecule

$$u_e(r) = \begin{cases} e^{-(x^2+y^2+z^2)} & r \in \Omega^- \\ 0 & r \in \Omega^+ \end{cases}$$



N	Newton iteration	$\|\nabla u - \nabla u_e\|_{\infty,\Gamma}$	order	$\|u - u_e\|_\infty$	order
10	4	6.572e-002	–	8.136e-003	–
20	3	1.378e-002	2.2538	2.025e-003	2.0064
40	3	3.115e-003	2.1292	4.901e-004	2.0467

Summary of computing Poisson-Boltzmann equation

- Ingredients: CIM + AMG + damped Newton’s iteration
- Capable to handle complex interfaces
- Second order accuracy for potential and electric field for molecules with smooth surfaces