Green's function method, pKa calculation, and Poisson-Boltzmann Equation

Jingzhen Hu

Contents

1 Introduction **2**

2 Green's function method **2**

2.1 Concepts and theory 2

2.2 Numerical implementation 6

2.3 Numerical results and discussion 6

3 Background on pKa **8**

3.1 Definition of pKa 8

3.2 Property of pKa 9

3.3 Measurement of pKa 9

4 Poisson-Boltzmann Equation (PBE) **10**

4.1 1D nonlinear PBE 11

4.1.1 Analytical solution 11

4.1.2 Numerical solution 14

4.1.3 Numerical solution: extend to a single problem, $[0, \infty]$ 21

4.2 1D solvent and protein system under PBE 22

4.2.1 Linear version 22

4.2.2 Nonlinear version: analytical solution 24

4.2.3 Nonlinear version: numerical solution 27

5 Conclusion **30**

Green's function method, pKa calculation, and Poisson-Boltzmann Equation

Jingzhen Hu

1 Introduction

This project is motivated by interest in computing the acid dissociation rate (pKa) at an amino acid titration site. In Southern Methodist University I worked under the instruction of Professor Weihua Geng on the topic of computing pKa. Then I came to University of Michigan to do the REU program with Professor Robert Krasny. I would like to thank Professor Robert Krasny and Professor Weihua Geng for the instructions on this project. First I learnt how to use the Green's function method to solve two-point boundary problems for differential equations. After that we focus on developing a numerical algorithm to solve nonlinear Poisson-Boltzmann equation (PBE). This report introduces the Green's function method, background of pKa, and the PBE.

2 Green's function method

The main idea about the section is to solve the general second order ordinary differential equation $Ay" + By' + Cy = r(x)$ on any domain $x \in$ [a,b] based on two-point boundary conditions $y(a) = \alpha$, $y(b) = \beta$ in numerical way. The process can be divided into two parts: one is to express the solution as an integral of the corresponding Green's function (possibly with some boundary terms) and the other is to use the numerical integration to evaluate the solution on uniform points. Finally, both the exact solution and the numerical solution are plotted into a graph.

2.1 Concepts and theory

In specific, the section will focus on the problem,

$$y'' + 2y' + y = -4e^x, \quad 0 \leq x \leq 1, \tag{1}$$

with boundary conditions,

$$y(0) = -1, \quad y(1) = 2 - e. \tag{2}$$

First, we solve it analytically to get the exact solution. Using the characteristic equation we get the homogeneous solution $y_h = ae^{-x} + bxe^{-x}$ and the particular solution $y_p = ce^x$. After that, we get the $a = 0$, $b = 2$e, $c = -1$ which leads to the exact solution $y = 2xe^{1-x} - e^x$.

Next we find the solution based on the Green's function and compare the results. Initially introducing some definitions that are involved in the process [1]:

- $\mathbf{L}$ is an n^{th} order linear ordinary differential operator.

$$\mathbf{L} = a_n(x)\frac{d^n}{dx^n} + a_{n-1}(x)\frac{d^{n-1}}{dx^{n-1}} + ... + a_1(x)\frac{d}{dx} + a_0(x) \tag{3}$$

  In the example above we have

$$\mathbf{L} = A\frac{d^2}{dx^2} + B\frac{d}{dx} + C = \frac{d^2}{dx^2} + 2\frac{d}{dx} + 1. \tag{4}$$

- $\mathbf{L}^*$ is the formal adjoint differential operator and can be defined by the equation (using integration by parts),

$$\int_a^b v\mathbf{L}u dx = [\cdots]\Big|_a^b + \int_a^b u\mathbf{L}^* v dx. \tag{5}$$

  Generally if $\mathbf{L} = \mathbf{L}^*$, then $\mathbf{L}$ is formally self-adjoint.

- The delta function, $\delta(x)$, is a generalized function which has an infinite value at $x = 0$ and is zero in the rest of the domain, with the property

$$\int_{-\infty}^{\infty} \delta(x - \xi) f(x) = f(\xi). \tag{6}$$

- The delta function is the derivative of the Heaviside step function,

$$H'(x - \xi) = \delta(x - \xi), \tag{7}$$

  where $H(x)$ is defined by

$$H(x) = \begin{cases} 1 \ , \ x > 0 \\ 0 \ , \ x < 0 \end{cases} \tag{8}$$

- The Green's function, $G(\xi, x)$, is defined by the property

$$\mathbf{L}^* G(\xi, x) = \delta(\xi - x), \tag{9}$$

  where $\mathbf{L}^*$ acts with respect to ξ and x is a parameter. A homogeneous solution of the adjoint operator can be added to the Green's function.

With this mathematical setting, we can compute as follows,

$$\begin{aligned}\int_0^1 G(\xi,x)\mathbf{L}y(\xi)d\xi &= \int_0^1 G(\xi,x)(Ay''(\xi)+By'(\xi)+Cy(\xi))d\xi \\ &= [GAy' - G_\xi Ay + GBy]\Big|_{\xi=0}^{\xi=1} + \int_0^1 y(AG_{\xi\xi} - BG_\xi + CG)d\xi \\ &= [Gy' - G_\xi y + 2Gy]\Big|_{\xi=0}^{\xi=1} + \int_0^1 y(G_{\xi\xi} - 2G_\xi + G)d\xi \\ &= G(1,x)y'(1) - G_\xi(1,x)y(1) + 2G(1,x)y(1) \\ &- G(0,x)y'(0) + G_\xi(0,x)y(0) - 2G(0,x)y(0) + \int_0^1 y\mathbf{L}^*Gd\xi\end{aligned} \tag{10}$$

In order to eliminate the two unwelcome terms $y'(1)$ and $y'(0)$, we can choose two boundary conditions for the Green's function,

$$G(1,x) = G(0,x) = 0. \tag{11}$$

Since the boundary conditions are $y(0) = -1$ and $y(1) = 2 - e$, then we can simplify equation (10) using the property of the Green's function in equation (9),

$$\begin{aligned}\int_0^1 G\mathbf{L}yd\xi = \int_0^1 G(\xi,x)r(\xi)d\xi &= -y(1)G_\xi(1,x) + y(0)G_\xi(0,x) + \int_0^1 y\delta(\xi-x)d\xi \\ &= -(2-e)G_\xi(1,x) + (-1)G_\xi(0,x) + y(x).\end{aligned} \tag{12}$$

Then using $r(x) = -4e^x$, we obtain the solution,

$$y(x) = \int_0^1 G(\xi,x)(-4e^\xi)d\xi + (2-e)G_\xi(1,x) + G_\xi(0,x). \tag{13}$$

To obtain the Green's function, we use the property that $\delta(\xi - x) = 0$ when $\xi \neq x$. Then we solve the homogeneous ODE, $G_{\xi\xi} - 2G_\xi + G = 0$, in two intervals, $0 \leq \xi < x$ and $x < \xi \leq 1$, in the following expression,

$$G(\xi,x) = \begin{cases} Ae^\xi + B\xi e^\xi & 0 \leq \xi < x \\ Ce^\xi + D\xi e^\xi & x < \xi \leq 1. \end{cases} \tag{14}$$

To figure out the coefficients A, B, C, D, we already have the two general boundary conditions of the Green's function,

$$G(0,x) = A = 0, \quad G(1,x) = C + D = 0. \tag{15}$$

Besides, we need two other conditions. Integrating the $\mathbf{L}^*G = G_{\xi\xi} - 2G_\xi + G = \delta(\xi - x)$ from $x - 0$ to $x + 0$, we can get them

$$\int_{x-0}^{x+0} (G_{\xi\xi} - 2G_\xi + G)d\xi = \int_{x-0}^{x+0} \delta(\xi - x)d\xi, \tag{16}$$

$$G_\xi\Big|_{\xi=x-0}^{\xi=x+0} - 2G\Big|_{\xi=x-0}^{\xi=x+0} + \int_{x-0}^{x+0} Gd\xi = 1. \tag{17}$$

The second and third term approach to zeros if the $G(\xi, x)$ is a continuous function of ξ at $\xi = x$. Then, the left term forms a jump condition for the G_ξ

$$G_\xi\Big|_{\xi=x-0}^{\xi=x+0} = 1 \tag{18}$$

$$G_\xi(\xi, x) = \begin{cases} Ae^\xi + Be^\xi + B\xi e^\xi & 0 \le \xi < x \\ Ce^\xi + Ce^\xi + D\xi e^\xi & x < \xi \le 1 \end{cases} \tag{19}$$

It is because the discontinuity of the G_ξ, the $G_{\xi\xi}$ could derive a delta function which shows priori. Then apply the continuity of $G(\xi, x)$ and the jump condition, we can get the rest of the conditions working for the coefficients

$$A + Bx - C - Dx = 0 \tag{20}$$

$$-Ae^x - Be^x - Bxe^x + Ce^x + De^x + Dxe^x = 1 \tag{21}$$

Based on the equations (15), (20), (21), we can solve for A, B, C, D and get the Green's function as following,

$$G(\xi, x) = \begin{cases} (x-1)e^{-x}\xi e^\xi & 0 \le \xi < x \\ -xe^{-x}e^\xi + xe^{-x}\xi e^\xi & x < \xi \le 1 \end{cases} \tag{22}$$

Then we calculate the $G_\xi(\xi, x)$ in the expression,

$$G_\xi(\xi, x) = \begin{cases} (x-1)e^{-x}e^\xi + (x-1)e^{-x}\xi e^\xi & 0 \le \xi < x \\ -xe^{-x}e^\xi + xe^{-x}e^\xi + xe^{-x}\xi e^\xi & x < \xi \le 1 \end{cases} \tag{23}$$

Next we use the numerical midpoint integration method to evaluate the solution. Applying the midpoint integration via evaluating a matrix product of which the cost of operations is $O(n^2)$. Normally comparing with the operations cost of $O(n^3)$ from finite-difference method which needs to do the LU factorization, the Green's function method is more efficient for general cases.

2.2 Numerical implementation

We will compare two methods, the mid-point method and trapezoid method. For the midpoint value evaluation, discretizing the x variable by $x_i = ih, i = 0 : n$, where $h = (b-a)/n = 1/n$. Let $\xi_j, j = 1 : n$ be the mid-point of the j^{th} interval. Then we do the approximation of the equation (13)

$$u(x_i) = (2-e)G_\xi(1,x_i) + G_\xi(0,x_i) + \sum_{j=1}^{n} hG(\xi_j, x_i)r(\xi_j), \ \ i = 1 : n-1. \tag{24}$$

For the trapezoid evaluation, since the whole interval is $a \le x \le b$, discretizing the grids points into endpoints x_i, choose an integer n which $n \ge 1$ to divide the domain into $(n+1)$ intervals with $(n+2)$ points: $h = (a-b)/(n+1)$, $x_i = 0+ih$ where $i = 0 : n+1$. For each interior points x_i, $i = 1 : n$, summing up the function value. Then, adding the boundary terms and the one half function value of the two endpoints at zero and one.

$$\begin{aligned} u(x_i) \approx [&\frac{1}{2}G(0,x_i)\phi(0) + G(\xi_1,x_i)\phi(\xi_1) + ... \\ &+ G(\xi_{n-1},x_i)\phi(\xi_{n-1}) + \frac{1}{2}G(1,x_i)\phi(1)] \cdot h + (2-e)G_\xi(1,\xi_j) + G_\xi(0,\xi_j) \end{aligned} \tag{25}$$

2.3 Numerical results and discussion

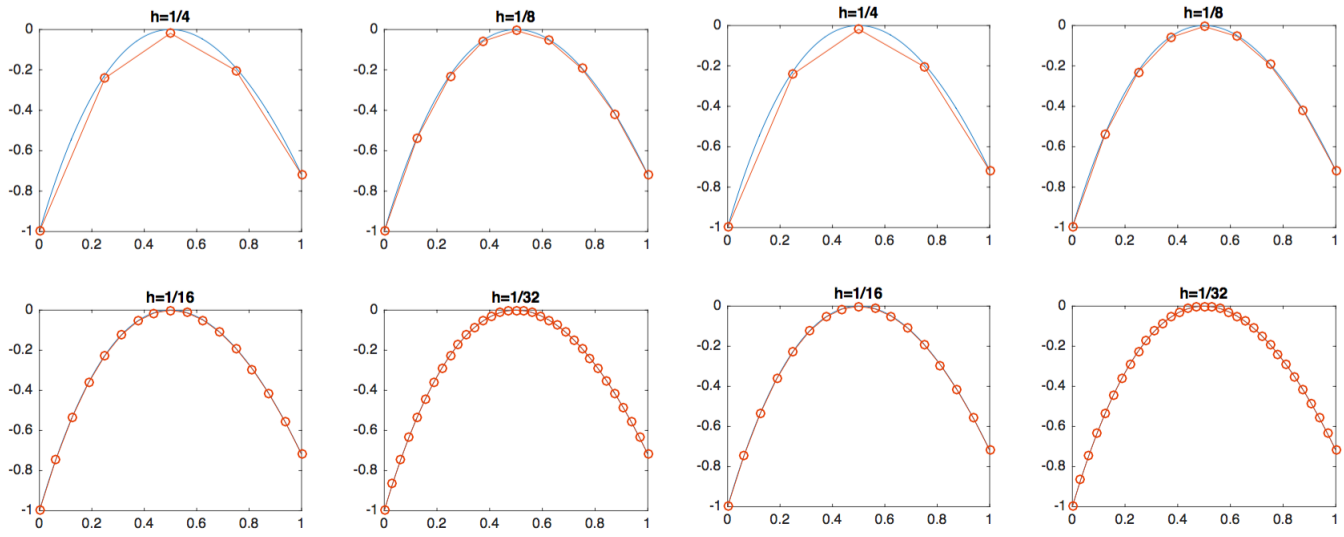

Figure 1: Green's function method, (left) midpoint method, (right) trapezoid method

In Figure 1, the origin circle is the polt of numerical solutions while the blue line is the exact

solutions on a fine mesh. Both solutions are polted in the same area in order to compare them clearly. With the decrease of the value of h, the value numerical solutions approximates the true solution better. We can see that using both numerical midpoint integration and trapezoid integration for Green's function method approximates the exact solution works out in a high accuracy. The results of the two integration methods match each other.

In Table 1, we can see the infinity norm of the error vector (value of true solution at united points subtract the numerical solution at the same points) decreases to the one fourth as the h decrease to the one half each time. As the h approach to zero, the norm of the error vector also will approximate to zero (the numerical value approach to the exact solution). In the process, the ratio of convergence(third column) will be steady to four and the last column also will be stable to two since the norm of the errors decays twice as the value of h. In other words, the forth column represents the order of the accuracy based on the value of h. The error analysis from mid-point method and trapezoid method match each other in seventh digit.

h	Max Error	Ratio of Convergence	Order of Precision
0.25000	0.0184230	0	0
0.12500	0.0046489	3.9628449	1.9865365
0.06250	0.0011729	3.9634054	1.9867405
0.03125	0.0002935	3.9959397	1.9985348
h	Max Error	Ratio of Convergence	Order of Precision
0.25000	0.0184230	0	0
0.12500	0.0046489	3.9628449	1.9865365
0.06250	0.0011729	3.9634054	1.9867405
0.03125	0.0002935	3.9959397	1.9985348

Table 1: Numerical results, (top) mid-point method, (bottom) trapezoid method.

As the project is motivated by a in-class project of solving the two point value boundary problem by finite-difference method with the direct LU factorization. Here add the results, Table 2, got in that project and find that even comparing with the result evaluated by the Green's function method, the errors of the finite-difference has a factor one-half more accuracy. I am somehow surprised with the result but I expected the Green's function method might works better in two or three dimensions problem.

h	Max Error	Ratio of Convergence	Order of Precision
0.25000	0.0096380	0	0
0.12500	0.0023772	4.0543458	2.0194691
0.06250	0.0005961	3.9876402	1.9955352
0.03125	0.0001486	4.0096749	2.0034852

Table 2: Numerical results, error from the direct LU factorization method

In this section, we solve the problem into two parts: first to apply the Green's function to the ODE and get the solution in the form of a integration plus the boundary terms; second to apply the numerical midpoint method and trapezoid method to approximate the integration. As in this case, the finite-difference method is more accurate then the Green's function evaluated by both the midpoint method and the Trapezoid rule. Although the operation cost of both the midpoint method and the Trapezoid rule is $O(n^2)$, comparing with the finite-difference method involving the direct LU factorization for a general dense matrix of which operation cost is $O(n^3)$, in some special case the finite-difference method could be better than the Green's function method and in others, it is not. Since it is a case-by-case discussion, I will not continue to make a simple conclusion here.

3 Background on pKa

3.1 Definition of pKa

The Ka, as an equilibrium constant for acid dissociation,

$$\mathrm{HA} + \mathrm{H_2O} \rightleftharpoons \mathrm{A^-} + \mathrm{H_3O^+}, \tag{26}$$

indicates the strength of ionization in an acid solution under specific temperature, is given by

$$\mathrm{Ka} = \frac{[\mathrm{A^-}][\mathrm{H_3O^+}]}{[\mathrm{HA}][\mathrm{H_2O}]}. \tag{27}$$

pKa, the logarithmic constant, measures the tendency for a group to give up a proton in the equation

$$\mathrm{pKa} = -\log_{10}\mathrm{Ka} = -\log_{10}\frac{[\mathrm{A^-}][\mathrm{H_3O^+}]}{[\mathrm{HA}]}. \tag{28}$$

3.2 Property of pKa

Since the negative sign, the value of pKa is inversely proportional to that of Ka. The smaller the value of pKa, the stronger the acid is. The relation between pKa and pH is given by the Henderson-Hasselbalch equation,

$$\mathrm{pH} = \mathrm{pKa} + \log_{10}\frac{[\mathrm{A}^-]}{[\mathrm{HA}]}. \tag{29}$$

Especially, at the half-equivalence-point, $[\mathrm{HA}] = [\mathrm{A}^-]$, the value of pKa equals to pH value.

3.3 Measurement of pKa

For simple-structured compound, recalling the Henderson-Hasselbalch Equation, equation (29), the pH value at the 50% titration site graph of acid-base reaction equals the pKa value. Since different types of amino acids and distinct structure of proteins make the pKa calculation more difficulty in proteins, the simple titration is difficult to control. Nuclear Magnetic Resonance spectra alternatively records the pH value of protein when a specific site is being protonated. However, the process needs enormous energies and money. Comparably, Poisson-Boltzmann (PB) solver, one of the computer simulation, becomes an effective and precise method to calculate the pKa by giving structure of proteins, electrostatics, and environmental parameters [2].

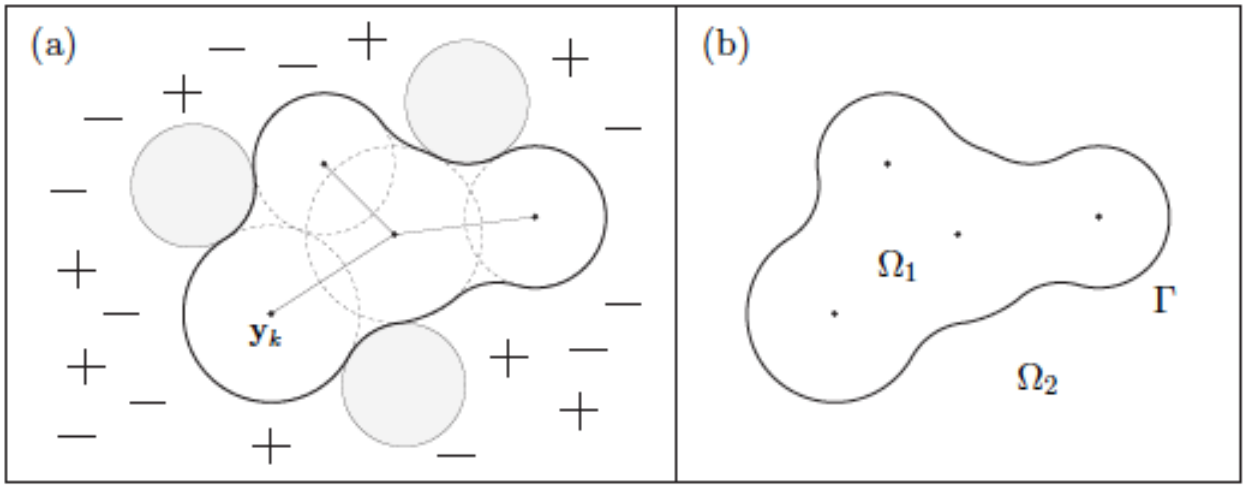

Figure 2: Protein-and-Solvent system[3]

The relation between the equilibrium constant, Ka, and standard Gibbs energy change, $\triangle G$,

for a reaction is given by

$$\triangle G = -\mathrm{RT}\ln \mathrm{Ka} \approx (2.303\mathrm{RT})\mathrm{pKa}, \tag{30}$$

where R = 8.31J/(mol·K). Since the pKa of the amino-acid alone is given by experiments, pKa of the amino-aid as a residue of a protein is determined by the difference of solvation energy($\triangle(\triangle G)$) under the electrostatic free energy. The Figure 2 is the numerical simulation to compute pKa. Ω_1 represents the protein part with the dielectric constant ϵ_p while Ω_2 is the solvent part with the ϵ_s. y_k is one of the site in the site on the protein with corresponding charge q_k. The $\triangle G$ is obtained by the electrostatic potential value and its derivative on the molecular surface Γ.

$\phi(\boldsymbol{x})$ is defined as the electrostatic potential in the protein-and-solvent system. Given the dielectric constants ϵ_1 in protein and ϵ_2 in solvent and the screening constant κ, the $\phi(\boldsymbol{x})$ can be considered as the solution of the Poisson-Boltzmann Equation as follows,

$$-\epsilon_p \nabla^2 \phi_1(\boldsymbol{x}) = \sum_{k=1}^{N} q_k \delta(\boldsymbol{x} - \boldsymbol{y_k}) \qquad \boldsymbol{x} \in \Omega_1 \tag{31}$$

$$-\epsilon_s \nabla^2 \phi_2(\boldsymbol{x}) + \sinh \kappa^2 \phi_2(\boldsymbol{x}) = 0 \qquad \boldsymbol{x} \in \Omega_2 \tag{32}$$

with the interface conditions(on Γ),

$$\phi_1(\boldsymbol{x}) = \phi_2(\boldsymbol{x}), \quad \epsilon_p \frac{\partial \phi_1(\boldsymbol{x})}{\partial V} = \epsilon_s \frac{\partial \phi_2(\boldsymbol{x})}{\partial V}. \tag{33}$$

4 Poisson-Boltzmann Equation (PBE)

The equation (31) and (32) represents the potential related to the charge density which is divided into internal constant charge of the solute (ρ_1),

$$\rho_1 = \sum_{k=1}^{N} q_k \delta(\boldsymbol{x} - \boldsymbol{y_k}), \tag{34}$$

and external distribution of ion in the solvent (ρ_2),

$$\rho_2 = \frac{e^{-\kappa^2 \phi(\boldsymbol{x})} - e^{\kappa^2 \phi(\boldsymbol{x})}}{2} = -\sinh \kappa^2 \phi(\boldsymbol{x}). \tag{35}$$

For solvent part, the experiment shows that the charge distribution in the medium follows the Boltzmann distribution. As the right term -sinh $\kappa^2 \phi(\boldsymbol{x})$ is a nonlinear term, it is better to linearize the sinh function in the Tayler Series and keep the first term $\kappa^2 \phi(\boldsymbol{x})$ when κ is a small value near zero,

$$-\epsilon_s \nabla^2 \phi(\boldsymbol{x}) + \kappa^2 \phi(\boldsymbol{x}) = 0 \qquad \boldsymbol{x} \in \Omega_s. \tag{36}$$

4.1 1D nonlinear PBE

The 1D Poisson-Boltzmann Equation to study of electrostatics in salty solutions is showed

$$\frac{d^2\phi(x)}{dx^2} = \frac{2nq}{\epsilon_0}\sinh\frac{q\phi(x)}{\kappa_B T}, \tag{37}$$

where ϕ is the electric potential, n is the ion density, q is the ion charge, T is the temperature, and ϵ_0 and κ_B are constants[4].

4.1.1 Analytical solution

Using change of variable to transfer the equation in dimensionless units. Define $\hat{x} = ax$, $\hat{\phi} = b\phi$, the equation (37) becomes

$$\frac{d\phi}{dx} = \frac{d(\frac{\hat{\phi}}{b})}{d\hat{x}} \cdot \frac{d\hat{x}}{dx} = \frac{a}{b} \cdot \frac{d\hat{\phi}}{d\hat{x}}, \tag{38}$$

$$\frac{d^2\phi}{dx^2} = \frac{d(d\phi/dx)}{dx} = \frac{d(\frac{a}{b}\hat{\phi}'(\hat{x}))}{dx} = \frac{d(\frac{a}{b}\hat{\phi}'(\hat{x}))}{d\hat{x}} \cdot \frac{d\hat{x}}{dx} = \frac{a^2}{b}\frac{d^2\hat{\phi}}{d\hat{x}^2}, \tag{39}$$

$$\phi''(x) = \frac{2nq}{\epsilon_0}\sinh\frac{q\phi(x)}{\kappa_B T}, \tag{40}$$

$$\Rightarrow \frac{a^2}{b}\hat{\phi}''(\hat{x}) = \frac{2nq}{\epsilon_0}\sinh\frac{q\hat{\phi}(\hat{x})}{\kappa_B T b}, \tag{41}$$

$$\Rightarrow \hat{\phi}''(\hat{x}) = \frac{2nqb}{\epsilon_0 a^2}\sinh\frac{q\hat{\phi}(\hat{x})}{\kappa_B T b}, \tag{42}$$

Choose a and b,

$$a = q\sqrt{\frac{2n}{\epsilon_0\kappa_B T}}, \quad b = \frac{q}{2\kappa_B T}, \tag{43}$$

$$\Rightarrow \hat{\phi}''(\hat{x}) = \frac{1}{2}\sinh 2\hat{\phi}(\hat{x}), \tag{44}$$

with $\hat{x} \in [0,\infty)$ and two boundary condition $\hat{\phi}(0) = \psi$, $\hat{\phi}(\infty) = 0$.

$$\hat{\phi}'' \cdot \hat{\phi}' = \frac{1}{2}\sinh 2\hat{\phi} \cdot \phi' \tag{45}$$

$$\frac{1}{2}(\hat{\phi}')^2 = \frac{1}{4}\cosh 2\hat{\phi} + A \tag{46}$$

Since $\hat{\phi}(\infty) = 0$, $A = -\frac{1}{4}$. Taking a square root both side leads to,

$$\hat{\phi}' = \sqrt{\frac{1}{2}\cosh 2\hat{\phi} - \frac{1}{2}} = \sqrt{\cosh^2\hat{\phi} - 1} = -\sinh\hat{\phi}. \tag{47}$$

In order to fit in the boundary condition, only keep the negative square root,

$$-\frac{d\hat{\phi}}{\sinh\hat{\phi}} = dx. \tag{48}$$

Let $u = \cosh\hat{\phi}$, then $du = \sinh\hat{\phi}\, d\hat{\phi}$. Left hand side can be simplified as

$$-\frac{d\hat{\phi}}{\sinh\hat{\phi}} = -\frac{\sinh\hat{\phi}\, d\hat{\phi}}{\sinh^2\hat{\phi}} = -\frac{\sinh\hat{\phi}\, d\hat{\phi}}{\cosh^2\hat{\phi} - 1} = \frac{du}{u^2 - 1}. \tag{49}$$

In order to apply the integration both side and turn into the inverse hyperbolic tangent function, we introduce a new variable $w = \frac{1}{u}$, it follows

$$\frac{du}{u^2 - 1} = -\frac{dw}{w^2(\frac{1}{w^2} - 1)} = \frac{dw}{w^2 - 1}. \tag{50}$$

Integration both side, we get

$$\tanh^{-1}(w) + B = \hat{x} \qquad where\ w = \frac{1}{\cosh\hat{\phi}}. \tag{51}$$

As $\hat{\phi}(0) = \psi$, then $B = -\tanh^{-1}(\frac{1}{\cosh\psi})$.

$$-\tanh^{-1}(\frac{1}{\cosh\psi}) + \tanh^{-1}(\frac{1}{\cosh\hat{\phi}}) = \hat{x}, \tag{52}$$

$$-\frac{1}{2}\ln(\frac{\cosh\psi + 1}{\cosh\psi - 1}) + \frac{1}{2}\ln(\frac{\cosh\hat{\phi} + 1}{\cosh\hat{\phi} - 1}) = \hat{x}, \tag{53}$$

$$(\frac{\cosh\hat{\phi} + 1}{\cosh\hat{\phi} - 1} \cdot \frac{\cosh\psi - 1}{\cosh\psi + 1})^{\frac{1}{2}} = e^{\hat{x}}. \tag{54}$$

Apply $\cosh v + 1 = 2cosh^2(v/2)$, $\cosh v - 1 = 2\sinh^2(v/2)$ to simplify the equation,

$$\frac{\tanh\frac{\psi}{2}}{\tanh\frac{\hat{\phi}}{2}} = e^{\hat{x}}. \tag{55}$$

The solution is

$$\hat{\phi}(\hat{x}) = 2\tanh^{-1}(\tanh(\frac{\psi}{2})e^{-\hat{x}}). \tag{56}$$

If choosing the domain of x to be $(-\infty, 0]$ and the boundary conditions $\phi(-\infty) = 0$, $\phi(0) = \psi$, then we will get the solution

$$\phi_{nl} = 2\tanh^{-1}(\tanh(\frac{\psi}{2})e^{x}). \tag{57}$$

Taking the Taylor Series of $2\tanh^{-1}(\theta)$ near $\theta = 0$ and keeping the first term, we get the behavior near infinity

$$\phi_{nl} \sim 2\tanh^{-1}(\frac{\psi}{2}) \cdot e^{-x} \quad as\ x \to \infty. \tag{58}$$

As for $x \to 0$, using the Taylor Series near $\theta = \tanh(\frac{\psi}{2})$,

$$\begin{aligned}\phi_{nl}(x) \approx \phi_{nl}(0) + \phi'_{nl}(0) \cdot (\tanh(\frac{\psi}{2})e^{-x} - \tanh(\frac{\psi}{2})) &= \psi - 2\cosh^2(\psi)\tanh(\psi)(1 - e^{-x}) \\ &\approx \psi - \sinh\psi(1 - (1 - x))\end{aligned} \tag{59}$$

It turns to be

$$\phi_{nl} \sim \psi - \sinh\psi x \quad as\ x \to 0 \tag{60}$$

To compare the solution of linearized PBE and non-linearized PBE, it is convenient to solve the linearized PBE $\phi'' = \phi$ under the same boundary conditions. The solution $\phi_l = \psi e^{-x}$ can be used as the initial guess for the numerical iterative method. In Figure 3, when the value

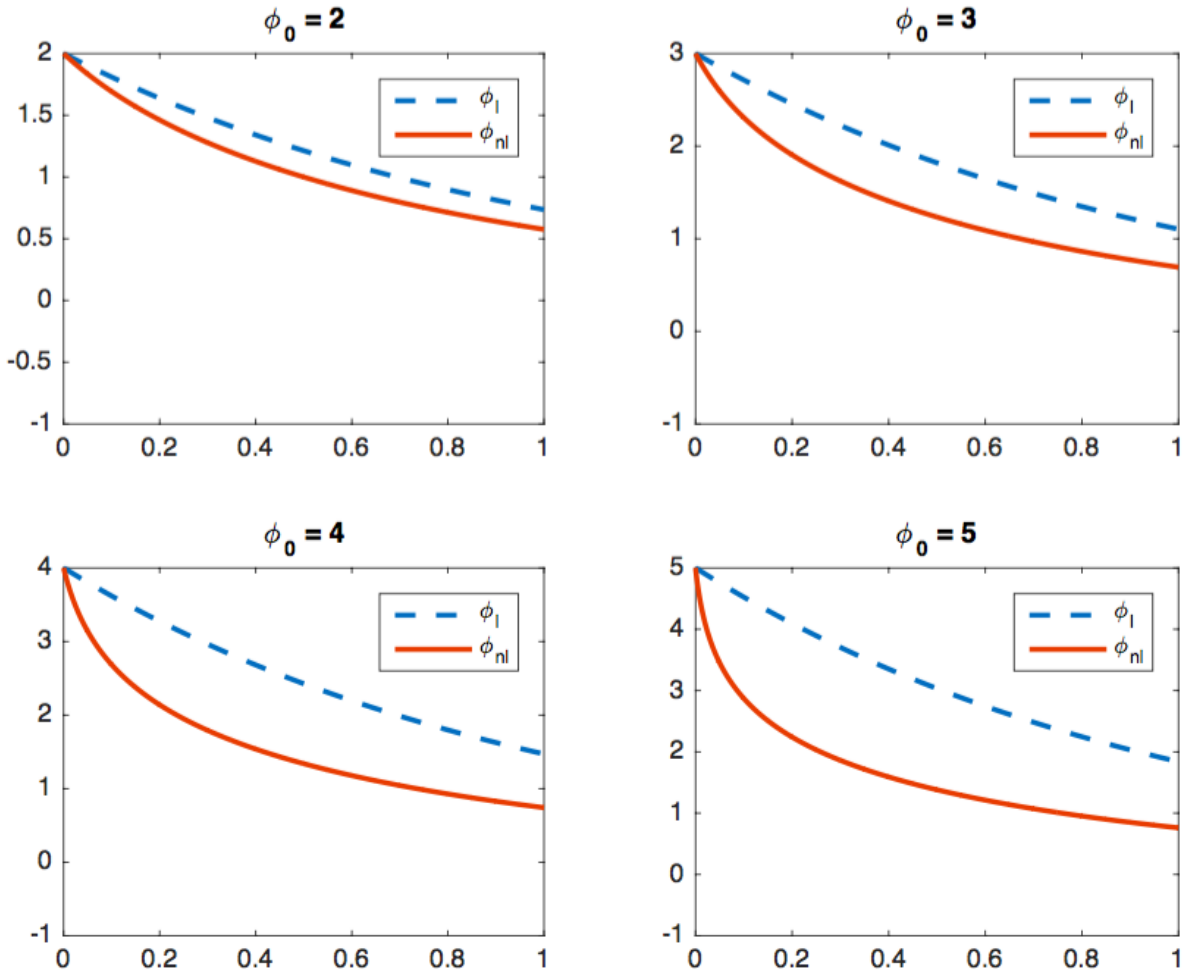

Figure 3: comparison of linear(dashed blue curve), nonlinear solution(red curve)

of ψ is small, the two solutions are very similar, but the nonlinear solution decays faster as increasing the value of ψ. However, when the salt concentration becomes larger in the solvent,

the differences between the linear and nonlinear solution turns to be lager. The actual decays faster if we use the linear solution. Therefore, it is necessary to study the nonlinear PBE under a more general numerical method.

4.1.2 Numerical solution

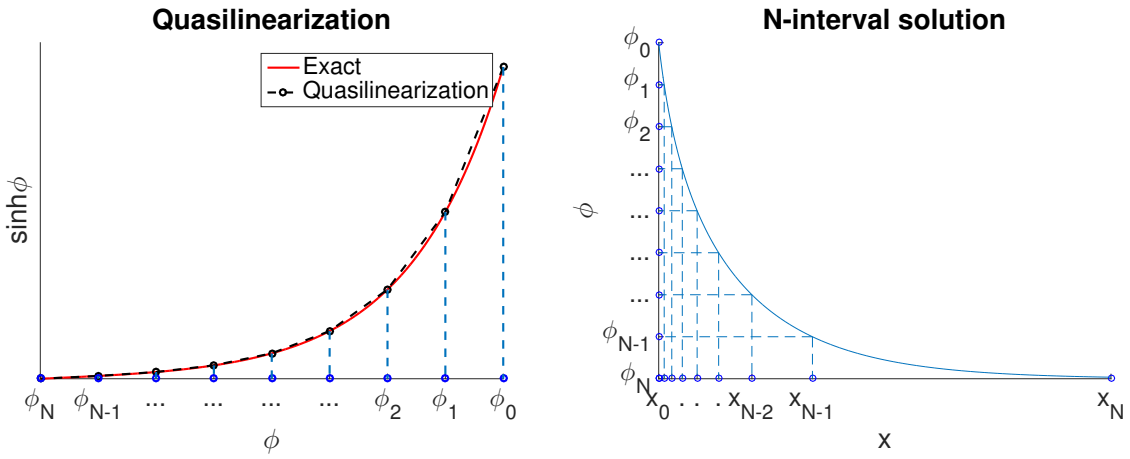

Figure 4: Discretization, uniform ϕ mesh points with corresponding projected x points

Since the process to get the analytic solution is uncertain and complex, a numerical way, quasi-linearization, is necessary to the solve nonlinear differential equations. In The quasilinearization technique employs a piecewise linear approximation of the nonlinear term in the equation. The mesh points are chosen by projecting uniform values of the potential onto the domain, and matching conditions are enforced at the interior mesh points. The Figure 4 gives a N-interval solution and its approximation of the right hand side function.

Before looking the single problem, we solve for a two point boundary value problem at first,

$$\phi''(x) = \frac{1}{2}\sinh(2\phi(x)), \tag{61}$$

$$\phi(0) = \psi, \quad \phi(L) = 2\tanh^{-1}(\tanh(\frac{\psi}{2})e^{-L}) \tag{62}$$

Let define the right hand side a function $f(\phi)$ to make the case more gerenal,

$$f(\phi) = \frac{1}{2}\sinh(2\phi). \tag{63}$$

Apply the idea of quasilinearization[5], we replace the right hand side function, $\frac{1}{2}sinh(2\phi)$, by its equal linear interpolation under a ϕ-mesh (ϕ_0, ϕ_1,..., ϕ_N) with corresponding x value (x_0,

x_1,...,x_N), where

$$0 = x_0 < x_1 < x_2 < ... < x_N = L. \tag{64}$$

Let us define the ith interval by $x_{i-1} \leq x \leq x_i$, for $i = 1 : N$. In ith interval, $i = 1 : N$, the equation (61) and (62) turn to be,

$$\phi_i''(x) = \alpha_i^2(\phi_i(x) - \phi_{i-1}) + f(\phi_{i-1}), \tag{65}$$

$$\phi_i(x_{i-1}) = \phi_{i-1},\ \phi_i(x_i) = \phi_i, \tag{66}$$

where the

$$\alpha_i^2 = \frac{f(\phi_i) - f(\phi_{i-1})}{\phi_i - \phi_{i-1}}. \tag{67}$$

Solving the equation using a combination of homogeneous solution and particular solution. The homogeneous solution satisfies the boundary condition (66). The particular solution satisfy the zero boundary condition. First, find a combination of the fundamental solutions, $u_i(x)$ and $v_i(x)$ that satisfy the boundary condition,

$$u_i(x_{i-1}) = v_i(x_i) = 0, \qquad u_i(x_i) = v_i(x_{i-1}) = 1. \tag{68}$$

The $u_i(x)$ and $v_i(x)$ is given by,

$$u_i(x) = \frac{\sinh \alpha_i(x - x_{i-1})}{\sinh \alpha_i(x_i - x_{i-1})}, \quad v_i(x) = \frac{\sinh \alpha_i(x_i - x)}{\sinh \alpha_i(x_i - x_{i-1})}. \tag{69}$$

The solution is in the form,

$$\phi_i(x) = \phi_i u_i(x) + \phi_{i-1} v_i(x) + P_i(x). \tag{70}$$

where the $P_i(x)$ is the particular solution which satisfies the $P_i(x_{i-1}) = P_i(x_i) = 0$. Since the Green's function exists for the problem, the particular solution can be expressed into[5],

$$P_i(x) = \frac{v_i(x)}{c_i} \int_{x_{i-1}}^{x} u_i(\xi)(f(\phi_{i-1}) - \alpha_i^2\phi_{i-1})d\xi + \frac{u_i(x)}{c_i} \int_{x}^{x_i} v_i(\xi)(f(\phi_{i-1}) - \alpha_i^2\phi_{i-1})d\xi \tag{71}$$

where

$$c_i = \begin{vmatrix} u_i & v_i \\ u_i' & v_i' \end{vmatrix} = u_i v_i' - u_i' v_i = -\frac{\alpha_i}{\sinh \alpha_i(x_{i+1} - x_i)}. \tag{72}$$

After simplification, the particular solution is,

$$P_i(x) = (\phi_{i-1} - \frac{f(\phi_{i-1})}{\alpha_i^2})(1 - u_i(x) - v_i(x)). \tag{73}$$

Define a new variable,

$$\beta_i = \frac{f(\phi_{i-1})}{\alpha_i^2}, \tag{74}$$

to make equations cleaner. Recalling the equation (70), the solution is,

$$\phi_i(x) = \phi_i u_i(x) + \phi_{i-1} v_i(x) + (\phi_{i-1} - \beta_i)(1 - u_i(x) - v_i(x)) \tag{75}$$

Next, we patch up all the piecewise solutions by setting the derivative of the solutions match at the interiors,

$$\phi_i'(x_i) = \phi_{i+1}'(x_i), \quad i = 1 : N-1. \tag{76}$$

The derivative form of the solution is,

$$\phi_i'(x) = \phi_i u_i'(x) + \phi_{i-1} v_i'(x) - (\phi_{i-1} - \beta_i)(u_i'(x) + v_i'(x)) \tag{77}$$

where

$$u_i'(x) = \frac{\alpha_i \cosh \alpha_i(x - x_{i-1})}{\sinh \alpha_i(x_i - x_{i-1})}, \quad v_i'(x) = -\frac{\alpha_i \cosh \alpha_i(x_i - x)}{\sinh \alpha_i(x_i - x_{i-1})}. \tag{78}$$

The boundary for derivative of $u_i(x)$ and $v_i(x)$ is,

$$u_i'(x_{i-1}) = \frac{\alpha_i}{\sinh \alpha_i(x_i - x_{i-1})}, \quad u_i'(x_i) = \frac{\alpha_i}{\tanh \alpha_i(x_i - x_{i-1})} \tag{79}$$

$$v_i'(x_{i-1}) = -\frac{\alpha_i}{\tanh \alpha_i(x_i - x_{i-1})}, \quad v_i'(x_i) = -\frac{\alpha_i}{\sinh \alpha_i(x_i - x_{i-1})} \tag{80}$$

For the x_i point, the equation (76), which is expressed by

$$\begin{aligned} &\phi_i u_i'(x_i) + \phi_{i-1} v_i'(x_i) - (\phi_{i-1} - \beta_i)(u_i'(x_i) + v_i'(x_i)) = \\ &\qquad \phi_{i+1} u_{i+1}'(x_i) + \phi_i v_{i+1}'(x_i) - (\phi_i - \beta_{i+1})(u_{i+1}'(x_i) + v_{i+1}'(x_i)). \end{aligned} \tag{81}$$

Simplify the patching condition further,

$$\begin{aligned} &\phi_{i-1} v_i'(x_i) + \phi_i(u_i'(x_i) - v_{i+1}'(x_i)) - \phi_{i+1} u_{i+1}'(x_i) = \\ &\qquad (\phi_{i-1} - \beta_i)(u_i'(x_i) + v_i'(x_i)) - (\phi_i - \beta_{i+1})(u_{i+1}'(x_i) + v_{i+1}'(x_i)). \end{aligned} \tag{82}$$

To form a row of a matrix, we cleaning the equation to a form,

$$a_i \phi_{i-1} + b_i \phi_i + c_i \phi_{i+1} = rhs_i, \qquad i = 1, 2, ..., N-1 \tag{83}$$

where

$$a_i = v_i'(x_i), \quad b_i = u_i'(x_i) - v_{i+1}'(x_i), \quad c_i = -u_{i+1}'(x_i), \tag{84}$$

$$rhs_i = (\phi_{i-1} - \beta_i)(u_i'(x_i) + v_i'(x_i)) - (\phi_i - \beta_{i+1})(u_{i+1}'(x_i) + v_{i+1}'(x_i)), \tag{85}$$

where A is a tridiagonal and symmetric matrix.

Combining all the patching condition, the equation (83), we can form them into a matrix,

$$A(x, \phi)\phi = rhs(x, \phi), \tag{86}$$

$$\begin{pmatrix} b_1 & c_1 & 0 & 0 & \cdots & 0 \\ 0 & a_2 & b_2 & c_2 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \cdots & 0 \\ 0 & \cdots & 0 & a_{N-2} & b_{N-2} & c_{N-2} \\ 0 & \cdots & \cdots & 0 & a_{N-1} & b_{N-1} \end{pmatrix} \times \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{N-2} \\ \phi_{N-1} \end{pmatrix} = \begin{pmatrix} r_1 - a_1\phi(0) \\ r_2 \\ \vdots \\ r_{N-2} \\ r_{N-1} - c_{N-1}\phi(L) \end{pmatrix}. \tag{87}$$

If we choose a uniform mesh for ϕ, the error is bound by $O(N^{-2})$ [5]. To make sure the ϕ points are uniformly distributed in each iteration, we have an **inner** iteration with index k and an **outer** iteration with index j. The **inner** iteration is to update ϕ values with fixed x values to satisfy the patching condition,

$$A(x^j, \phi^k)\phi^{k+1} = rhs(x^j, \phi^k). \tag{88}$$

Given a convergence critera $\epsilon_\phi = 10^{-8}$, the inner iteration stop when

$$|\phi^{k+1} - \phi^k| < \epsilon_\phi. \tag{89}$$

Since the result of the **inner** iteration, the ϕ value will not be uniformly distributed. To reach the optimal error, we project the evenly distributed ϕ value on to the latest numerical solution that formed by the updated ϕ value obtained by the **inner** iteration to get new x points. Recalling the equation (75), we solve for the x value when we set the right hand side a uniform mesh of ϕ,

$$\phi^{k+1}(x^{j+1}) = \phi_{evenly}, \tag{90}$$

where ϕ_{evenly} is a uniform mesh of ϕ with N intervals. The i^{th} item in the ϕ_{evenly} is given by,

$$\phi^n_{evenly} = n\frac{\phi(0) - \phi(L)}{N}, \quad n = 1 : N - 1. \tag{91}$$

This is the **outer** iteration with a converge critera ϵ_x. The whole iteration stop when,

$$|x^{j+1} - x^j| < \epsilon_x. \tag{92}$$

Each time at the end of the **outer** iteration, we can multiple the number of intervals, N, by 2. The result turns out to be $N = 1, 2, 4, ..., 2^m$. For each iteration, the initial guess come form the previous solution. For example, the initial guess for the two intervals iteration comes form the result of the one interval solution. The first initial guess can be obtained with the result of one interval which is the yellow line in Figure 3. We project new equispaced ϕ points onto x-axis to obtain the initial guess for x point. Then, we go into the **inner** iteration to updated ϕ values and then the *outer* iteration to update x values.

A full expression of the equation (90),

$$\phi_i u_i(x) + \phi_{i-1} v_i(x) + (\phi_{i-1} - \beta_i)(1 - u_i(x) - v_i(x)) = \phi^n_{evenly}, \tag{93}$$

can be analytically for updated x value. Since only u_i and v_i contains the unknown x, we separate them on one side,

$$\beta_i v_i(x) + (\beta_i + \phi_i - \phi_{i-1}) u_i(x) = \phi^n_{evenly} - \phi_{i-1} + \beta_i. \tag{94}$$

To express the equation in a form only with constants and the unknown x, we define γ as follows,

$$\gamma = \sinh \alpha_i (x_i - x_{i-1}). \tag{95}$$

The equation (93) can be simplified as,

$$\frac{\beta_i \sinh \alpha_i (x_i - x) + (\beta_i + \phi_i - \phi_{i-1}) \sinh \alpha_i (x - x_{i-1})}{\gamma} = \phi^n_{evenly} - \phi_{i-1} + \beta_i. \tag{96}$$

Substitute the equation,

$$\begin{aligned} \sinh \alpha_i (x_i - x) &= \sinh(\alpha_i (x_i - x_{i-1}) - \alpha_i (x - x_{i-1}) \\ &= \sinh \alpha_i (x_i - x_{i-1}) \cosh \alpha_i (x - x_{i-1}) + \cosh \alpha_i (x_i - x_{i-1}) \sinh \alpha_i (x - x_{i-1}), \end{aligned} \tag{97}$$

and let $\omega = \cosh \alpha_i (x_i - x_{i-1})$, the equation turns to be,

$$\beta_i (\gamma \cosh \alpha_i (x - x_{i-1}) - \omega \sinh \alpha_i (x - x_{i-1})) + (\beta_i + \phi_i - \phi_{i-1}) \sinh \alpha_i (x - x_{i-1}) = \gamma (\phi^n_{evenly} - \phi_{i-1} + \beta_i). \tag{98}$$

Assume $X = \sinh \alpha_i (x - x_{i-1})$, it is further simplified as,

$$\gamma \beta_i \sqrt{1 + X^2} = (\omega \beta_i - \beta_i - \phi_i + \phi_{i-1}) X + \gamma (\phi^n_{evenly} - \phi_{i-1} + \beta_i). \tag{99}$$

Square each side,

$$\begin{aligned} \gamma^2 \beta_i^2 (1 + X^2) = (\omega \beta_i - \beta_i - \phi_i + \phi_{i-1})^2 X^2 + \\ 2(\omega \beta_i - \beta_i - \phi_i + \phi_{i-1}) \gamma (\phi^n_{evenly} - \phi_{i-1} + \beta_i) X + \gamma^2 (\phi^n_{evenly} - \phi_{i-1} + \beta_i)^2. \end{aligned} \tag{100}$$

Convert it into the quadratic form of x,

$$ax^2 + bx + c = 0, \tag{101}$$

where

$$a = \gamma^2\beta_i^2 - (\omega\beta_i - \beta_i - \phi_i + \phi_{i-1})^2 \tag{102}$$

$$b = -2(\omega\beta_i - \beta_i - \phi_i + \phi_{i-1})\gamma(\phi^n_{evenly} - \phi_{i-1} + \beta_i) \tag{103}$$

$$c = \gamma^2\beta_i^2 - \gamma^2(\phi^n_{evenly} - \phi_{i-1} + \beta_i)^2 \tag{104}$$

Apply the quadratic formula,

$$X = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad \text{or} \quad \mathrm{X} = \frac{2\mathrm{c}}{-\mathrm{b} \pm \sqrt{\mathrm{b}^2 - 4\mathrm{ac}}}, \tag{105}$$

depend on the sign of b to avoid the loss of significance. Then, the solution is in form of,

$$x = x_{i-1} + \frac{1}{\alpha_i}\sinh^{-1} X. \tag{106}$$

Since it seems to have two solution, we need to judge which one of them is in the right interval $[x_{i-1}, x_i]$.

The Figure 5 provides the pseudocode to implement the algorithm. The Figure 6 shows how the numerical solutions gradually approach the exact solution up to the eight-interval solution. The graph contains a original plot of solutions and a magnification version of the plot near the middle point of x.

N	Max Error	Max Error$\cdot$ N^2
2	0.01304870	0.05219483
4	0.00336140	0.05378243
8	0.00087613	0.05607263
16	0.00021902	0.05606993
32	0.00005475	0.05607271
64	0.00001370	0.05612182

Table 3: Numerical results: errors analysis for 2^k number of intervals, $\mathrm{k} = 1 : 6$

The Table 3 is the Error Analysis under the Err tolerance equals 1e-12. The first two columns are the number of intervals and the max error based on solution of such that number of intervals.

1 Provide the boundary information, x_0, x_N, ϕ_0, ϕ_N. Plot the exact solution. Let $N = 1$.

2 Plot the N-interval linear solution.

3 Project the uniformly distributed ϕ-mid point(s), ϕ_{mid}, onto the N-interval solution to get the x-mid, which is the initial guess for the inner iteration of $2N$-interval solution. For size(ϕ_{mid}) ≥ 3, we need first loop the number of ϕ_{mid} points. For each ϕ_{mid}, a nested loop for N intervals and judge which interval the ϕ_{mid} point is in.

Note that, we skip the even point in projection, since it should stay the same as the result of previous N-interval solution.

4 Do the inner iteration, using the matching condition to update ϕ value with fixed x value. The matching condition applies the iterative method, the equation (88).

5 Update the x value by projecting the ϕ_{mid} onto the current 2N-interval solution. The 2N-interval solution is generated by the current x value and updated phi value from inner iteration.

6 Go back to the inner iteration (step 4), and repeat doing the step 4 and 5 until the x value converges. This is the outer iteration.

7 Update the N value to be $2N$, $N = 2N$.

Figure 5: pseudocode for the quasilinearization technique

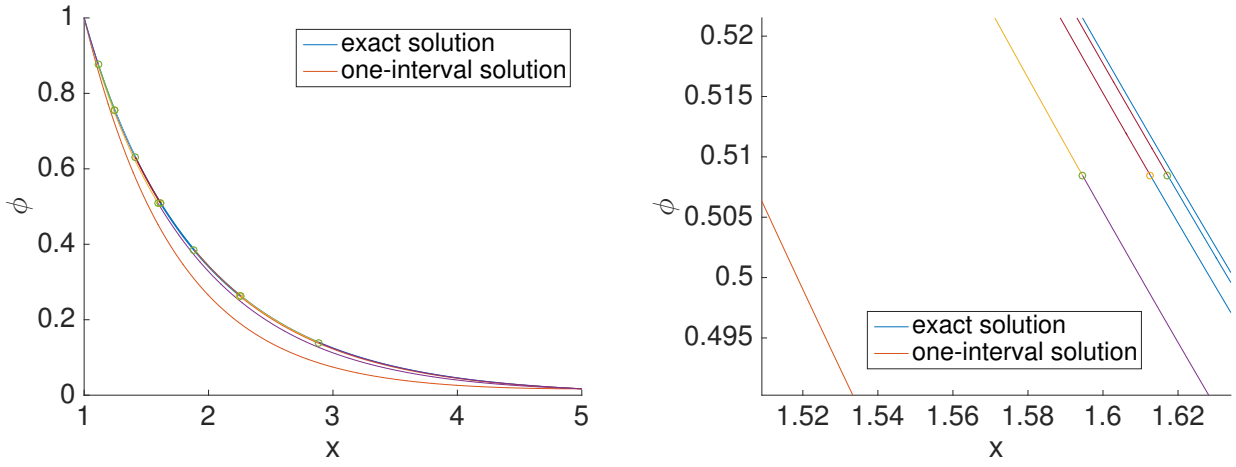

Figure 6: Numerical result: 8-Interval solution (left), its magnification around $\phi = 0{,}5$ (right)

The max error is calculated by the infinity norm of the difference vector of numerical results and exact solution on the mesh points. The third column indicates that the error is bound by $O(N^{-2})$. Note that each time increasing the number of interval, the $2N$-solution, which is closer to the exact solution, will be above the N-solution. Here is a simple proof.

With the same boundary condition, we define the difference of the two solution,

$$u = \phi(x; N, \phi_0) - \phi(x; 2N, \phi_0), \quad u(0) = 0, \quad u(\infty) = 0 \tag{107}$$

Then, the second derivative of u is given,

$$u'' = \phi''(x; N, \phi_0) - \phi''(x; 2N, \phi_0) \geq 0 \tag{108}$$

With zero boundary condition and positive second derivative, the function $u(x)$ can be draw all below the x-axis, which means,

$$u \leq 0 \Rightarrow \phi(x; N, \phi_0) \leq \phi(x; 2N, \phi_0). \tag{109}$$

4.1.3 Numerical solution: extend to a single problem, $[0, \infty]$

Next, we treat the last interval as a special case to make the domain of x becomes [0,∞]. Then, the equation (66) for the last interval becomes,

$$\phi_N(x_{N-1}) = \phi_{N-1}, \quad \phi_N(\infty) = 0. \tag{110}$$

Based on the new boundary condition, we solve the equation (65),

$$\phi_N''(x) = \alpha_N^2(\phi_N(x) - \phi_{N-1}) + f(\phi_{N-1}), \tag{111}$$

for a special case solution. Then, we revise the corresponding **inner** iteration and **outer** iteration.

The fundamental solutions are $e^{\alpha_N x}$ and $e^{-\alpha_N x}$. Suppose the particular solution is a constant C, then substitute into the equation (65),

$$C = \phi_{N-1} - \beta_{N-1}. \tag{112}$$

Then, the solution is given by,

$$\phi_N(x) = Ae^{\alpha_N x} + Be^{-\alpha_N x} + \phi_{N-1} - \beta_{N-1}. \tag{113}$$

Solving coefficients A and B to satisfy the boundary conditions in equation (110),

$$\phi_N(\infty) = 0 \Longrightarrow A = 0, \quad \phi_{N-1} - \beta_{N-1} = 0, \tag{114a}$$

$$\phi_N(x_{N-1}) = \phi_{N-1} \Longrightarrow B = \phi_{N-1}e^{\alpha_N x_{N-1}}. \tag{114b}$$

The solution, equation (113), and its derivative are as follows,

$$\phi_N(x) = \phi_{N-1}e^{-\alpha_N(x-x_{N-1})}, \quad \phi'_N(x) = -\alpha_N\phi_{N-1}e^{-\alpha_N(x-x_{N-1})}. \tag{115}$$

For the patching condition, **inner** iteration, the special case of the last interval turns to be,

$$\phi'_{N-1}(x_{N-1}) = \phi'_N(x_{N-1}). \tag{116}$$

Apply the equation (77) and (115) to expand the equation as follows,

$$\phi_{N-1}u'_{N-1}(x_{N-1}) - \phi_{N-2}u'_{N-1}(x_{N-1}) + \beta_{N-1}(u'_{N-1}(x_{N-1}) + v'_{N-1}(x_{N-1})) = -\alpha_N\phi_{N-1}. \tag{117}$$

Then last row of the matrix that applies the patching condition is,

$$-\phi_{N-2}u'_{N-1}(x_{N-1}) + \phi_{N-1}(u'_{N-1}(x_{N-1}) + \alpha_N) = -\beta_{N-1}(u'_{N-1}(x_{N-1}) + v'_{N-1}(x_{N-1})). \tag{118}$$

For the **outer** iteration, the process is quite direct,

$$\phi_{N-1}e^{-\alpha_N(x-x_{N-1})} = \phi_{evenly} \Rightarrow x = x_{N-1} + \frac{1}{\alpha_N}\ln\frac{\phi_{N-1}}{\phi_{evenly}}. \tag{119}$$

Very similar process for the domain of $(-\infty, 0]$ to treat the first interval as a special case,

$$\phi_1(x) = \phi_1 e^{\alpha_1(x-x_1)}, \quad \phi'_1(x) = \alpha_1\phi_1 e^{\alpha_1(x-x_1)}. \tag{120}$$

The **inner** iteration turns to be,

$$(u'_2(x_1) + \alpha_1)\phi_1 - u'_2(x_1)\phi_2 = \beta_2(u'_2(x_1) + v'_2(x_1)) \tag{121}$$

The **outer** iteration, it is,

$$\phi_1 e^{\alpha_N(x-x_1)} = \phi_{evenly} \Rightarrow x = x_1 + \frac{1}{\alpha_1}\ln\frac{\phi_{evenly}}{\phi_1}. \tag{122}$$

4.2 1D solvent and protein system under PBE

4.2.1 Linear version

In 1D, the equation (31) and (32) can be linearized as

$$-\epsilon_p\phi''(x) = q_m\delta(x-y) \qquad 0 \le x \le 1, 0 \le y \le 1, \tag{123a}$$

$$-\epsilon_s\phi''(x) + \kappa^2\phi(x) = 0 \qquad x \le 0 \text{ or } x \ge 1, \tag{123b}$$

with the boundary condition that when x approaches to $\pm\infty$, the value of ϕ approaches to zero. The corresponding matching condition is,

$$\phi(0^-) = \phi(0^+), \quad \phi(1^-) = \phi(1^+), \tag{124a}$$

$$\epsilon_s\phi'(0^-) = \epsilon_p\phi'(0^+), \quad \epsilon_p\phi'(1^-) = \epsilon_s\phi'(1^+). \tag{124b}$$

For the middle interval, the ϕ_2 is obtained by integration twice,

$$\phi_2(x) = -\frac{1}{\epsilon_p} \cdot [qH(x-y)(x-y) + Ax + B], \tag{125}$$

where A and B are two constants that will be determined by the matching condition later. For the two end intervals, let

$$\alpha = \frac{\kappa}{\sqrt{\epsilon_s}}. \tag{126}$$

Applying the boundary condition to each case, the ϕ_1 and ϕ_3 are in the form of

$$\phi_1(x) = Ce^{\alpha x} \quad \text{x} \leq 0, \tag{127}$$

$$\phi_3(x) = De^{-\alpha x} \quad \text{x} \geq 1. \tag{128}$$

Applying the matching conditions in equation (124a) and (124b), we get a linear system

$$B + \epsilon_p C = 0 \tag{129a}$$

$$A + \epsilon_s \alpha C = 0 \tag{129b}$$

$$A + B + \epsilon_p D e^{-\alpha} = -q(1-y) \tag{129c}$$

$$A - \epsilon_s \alpha D e^{-\alpha} = -q \tag{129d}$$

The final solution is,

$$\phi_(x) = \begin{cases} Ce^{\alpha x} & \text{x} \leq 0 \\ -\frac{1}{\epsilon_p}(Ax + B) & 0 \leq \text{x} \leq \text{y} \\ -\frac{1}{\epsilon_p}(q(x-y) + Ax + B) & \text{y} \leq \text{x} \leq 1 \\ De^{-\alpha x} & \text{x} \geq 1, \end{cases} \tag{130}$$

where,

$$C = \frac{q[\epsilon_p + \epsilon_s\alpha(1-y)]}{\epsilon_s\alpha(2\epsilon_p + \epsilon_s\alpha)}, \quad A = -\epsilon_s\alpha C, \quad B = -\epsilon_p C, \quad D = \frac{(A+q)e^{\alpha}}{\epsilon_s\alpha}. \tag{131}$$

The Figure 7 shows how the physical constant is related to solution. With a large κ value, the potential in the solvent will decrease a little bit slower. With a small difference of ϵ_s and ϵ_p,

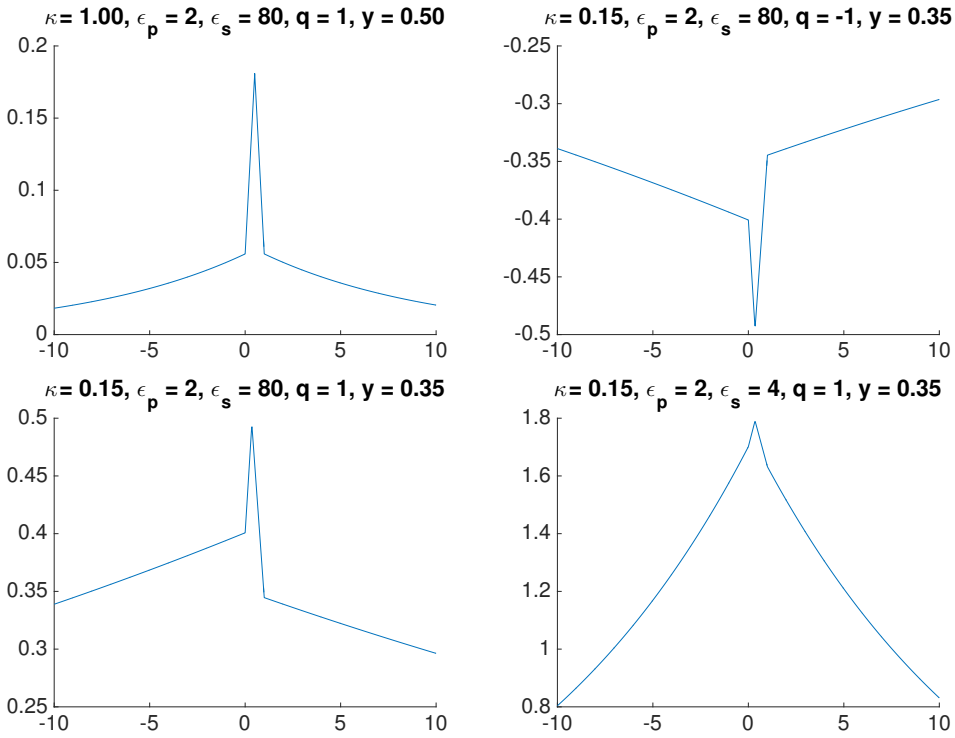

Figure 7: Analytic solution of linear protein-solvent system with different physical constants

the solution is smoother. The sign of charge can makehe solution revseral. The 1D nonlinear solvent and protein system is set as,

$$-\epsilon_p \phi''(x) = q_m \delta(x-y) \qquad 0 \le x \le 1, 0 \le y \le 1, \tag{132a}$$

$$\epsilon_s \phi''(x) = \sinh \kappa^2 \phi(x) \qquad x \le 0 \text{ or } x \ge 1, \tag{132b}$$

with the boundary condition that when x approaches to $\pm\infty$, the value of ϕ approaches to zero. The corresponding matching condition is same as the linear version,

$$\phi(0^-) = \phi(0^+), \quad \phi(1^-) = \phi(1^+), \tag{133a}$$

$$\epsilon_s \phi'(0^-) = \epsilon_p \phi'(0^+), \quad \epsilon_p \phi'(1^-) = \epsilon_s \phi'(1^+). \tag{133b}$$

4.2.2 Nonlinear version: analytical solution

For the two side intervals, following the drivation in section (4.1.1), assume the boundary condition,

$$\phi_1(0) = \phi_N^-, \quad \phi_3(1) = \phi_0^+. \tag{134}$$

Similarly, using change of variable, $\hat{x} = ax$, $\hat{\phi}(\hat{x}) = b\phi(x)$,

$$\frac{d\phi}{dx} = \frac{d(\frac{\hat{\phi}}{b})}{d\hat{x}} \cdot \frac{d\hat{x}}{dx} = \frac{a}{b} \cdot \frac{d\hat{\phi}}{d\hat{x}}, \tag{135}$$

$$\frac{d^2\phi}{dx^2} = \frac{d(d\phi/dx)}{dx} = \frac{d(\frac{a}{b}\hat{\phi}'(\hat{x}))}{dx} = \frac{d(\frac{a}{b}\hat{\phi}'(\hat{x}))}{d\hat{x}} \cdot \frac{d\hat{x}}{dx} = \frac{a^2}{b}\frac{d^2\hat{\phi}}{d\hat{x}^2}, \tag{136}$$

$$\phi''(x) = \frac{1}{\epsilon_s}\sinh\kappa^2\phi(x), \tag{137}$$

$$\Rightarrow \frac{a^2}{b}\hat{\phi}''(\hat{x}) = \frac{1}{\epsilon_s}\sinh\frac{\kappa^2}{b}\hat{\phi}(\hat{x}), \tag{138}$$

$$\Rightarrow \hat{\phi}''(\hat{x}) = \frac{b}{a^2\epsilon_s}\sinh\frac{\kappa^2}{b}\hat{\phi}(\hat{x}), \tag{139}$$

Choose a and b,

$$a = \kappa\sqrt{\frac{1}{\epsilon_s}}, \quad b = \frac{\kappa^2}{2}, \tag{140}$$

$$\Rightarrow \hat{\phi}''(\hat{x}) = \frac{1}{2}\sinh 2\hat{\phi}(\hat{x}). \tag{141}$$

Recalling the equation (57), when $x \in (-\infty, 0]$, the solution is,

$$\hat{\phi_1}(\hat{x}) = 2\tanh^{-1}(\tanh(\frac{\hat{\phi_N^-}}{2})e^{\hat{x}}) \Rightarrow b\phi_1(x) = 2\tanh^{-1}(\tanh(\frac{b}{2}\phi_N^-)e^{ax}). \tag{142}$$

Similarly, $x \in [1, \infty)$,

$$\hat{\phi_3}(\hat{x}) = 2\tanh^{-1}(\tanh(\frac{\hat{\phi_0^+}}{2})e^{-(\hat{x}-a)}) \Rightarrow b\phi_3(x) = 2\tanh^{-1}(\tanh(\frac{b}{2}\phi_0^+)e^{-a(x-1)}). \tag{143}$$

In order to apply the matching condition, we should compute $\phi_1'(0)$ and $\phi_3'(1)$ at first.

Deal with the equation (142),

$$\tanh(\frac{b}{2}\phi_1(x)) = \tanh(\frac{b}{2}\phi_N^-)e^{ax}. \tag{144}$$

Take derivative both sides,

$$\frac{b}{2}\operatorname{sech}^2(\frac{b}{2}\phi_1(x))\phi_1'(x) = \tanh(\frac{b}{2}\phi_N^-)e^{ax} \cdot a. \tag{145}$$

Let $x = 0$ and further simplify,

$$\phi_1'(0) = \frac{a}{b}\sinh(b\phi_N^-). \tag{146}$$

Deal with the equation (143),

$$\tanh(\frac{b}{2}\phi_3(x)) = \tanh^{-1}(\tanh(\frac{b}{2}\phi_0^+)e^{-a(x-1)}). \tag{147}$$

Take derivative both sides,

$$\frac{b}{2}\operatorname{sech}^2(\frac{b}{2}\phi_3(x))\phi_3'(x) = \tanh(\frac{b}{2}\phi_0^+)e^{-a(x-1)}\cdot(-a). \tag{148}$$

Let $x = 1$ and further simplify,

$$\phi_3'(1) = -\frac{a}{b}\sinh(b\phi_0^+). \tag{149}$$

For the middle interval, sloving the $\phi_2(x)$ using the homogeneous solution and particular solution. we can get the solution,

$$\phi_2(x) = \phi_0^+ x + \phi_N^-(1-x) - \frac{q}{\epsilon_p}(x-y)H(x-y) + \frac{q}{\epsilon_p}(1-y)x. \tag{150}$$

Its derivation,

$$\phi_2'(x) = \phi_0^+ - \phi_N^- - \frac{q}{\epsilon_p}H(x-y) + \frac{q}{\epsilon_p}(1-y), \tag{151}$$

at two endpoints,

$$\epsilon_p\phi_2'(0) = \epsilon_p(\phi_0^+ - \phi_N^-) + q(1-y), \quad \epsilon_p\phi_2'(1) = \epsilon_p(\phi_0^+ - \phi_N^-) - qy. \tag{152}$$

Then, we apply the matching condition to solve ψ_1, and ψ_3,

$$\frac{a}{b}\sinh(b\phi_N^-) = \epsilon_p(\phi_0^+ - \phi_N^-) + q(1-y), \tag{153a}$$

$$-\frac{a}{b}\sinh(b\phi_0^+) = \epsilon_p(\phi_0^+ - \phi_N^-) - qy, \tag{153b}$$

and use a fixed-point iterative method to solve the simplified system,

$$\epsilon_p\phi_N^- = -\frac{a}{b}\sinh(b\phi_N^-) + \epsilon_p\phi_0^+ + q(1-y), \tag{154a}$$

$$\epsilon_p\phi_0^+ = -\frac{a}{b}\sinh(b\phi_0^+) + \epsilon_p\phi_N^- + qy. \tag{154b}$$

With an index of iteration i, let the ϕ on the left side as the current step $(i+1)$ value and that on the right side as the previous step (i) value.

$$\epsilon_p\phi_{N,(i+1)}^- = -\frac{a}{b}\sinh(b\phi_{N,(i)}^-) + \epsilon_p\phi_{0,(i)}^+ + q(1-y), \tag{155a}$$

$$\epsilon_p\phi_{0,(i+1)}^+ = -\frac{a}{b}\sinh(b\phi_{0,(i)}^+) + \epsilon_p\phi_{N,(i+1)}^- + qy. \tag{155b}$$

Note that using the updated value $\phi_{N,(i+1)}^-$ at once in the next fixed-point iteration for $\phi_{0,(i+1)}^+$.

4.2.3 Nonlinear version: numerical solution

Apply the quasilinearization to the solvent part and add the matching condition at the molecular surface into the whole matching condition.

When the number of interval, $N = 1$, the matching condition at the 0 and 1 is treated as a special case. The result equals to the solution of the linear version. Recalling to equation (115), (120), (152), (133a) and (133b), we can derive,

$$\epsilon_s \alpha_1^- \phi_1^- = \epsilon_p(\phi_0^+ - \phi_1^-) + q(1-y) \tag{156a}$$

$$-\epsilon_s \alpha_1^+ \phi_0^+ = \epsilon_p(\phi_0^+ - \phi_1^-) - qy \tag{156b}$$

Write the two equations into a linear system, $A\phi = b$, where A is symmetric,

$$\begin{pmatrix} \epsilon_s\alpha_1^- + \epsilon_p & -\epsilon_p \\ -\epsilon_p & \epsilon_s\alpha_1^+ + \epsilon_p \end{pmatrix} \begin{pmatrix} \phi_1^- \\ \phi_0^+ \end{pmatrix} = \begin{pmatrix} q(1-y) \\ qy \end{pmatrix}. \tag{157}$$

As for $N \geq 2$, recalling the equation (77) the matching condition at the 0 and 1 turns to be,

$$\epsilon_s(\phi_N^- u_N^{'-}(0) + \phi_{N-1}^- v_N^{'-}(0) - (\phi_{N-1}^- - \beta_N^-)(u_N^{'-}(0) + v_N^{'-}(0))) = \epsilon_p(\phi_0^+ - \phi_N^-) + q(1-y) \tag{158}$$

$$\epsilon_s(\phi_1^+ u_1^{'+}(1) + \phi_0^+ v_1^{'+}(1) - (\phi_0^+ - \beta_0^+)(u_1^{'+}(1) + v_1^{'+}(1))) = \epsilon_p(\phi_0^+ - \phi_N^-) - qy \tag{159}$$

Put the two equations into the whole matching conditions. Since not involving normal middle matching condition, also treat $N = 2$ as a special case as follows,

$$\begin{pmatrix} u_2^{'-}(x_1^-) + \alpha_1^+ & -u_2^{'-}(x_1^-) & 0 & 0 \\ \epsilon_s v_N^{'-}(0) & \epsilon_s u_N^{'-}(0) + \epsilon_p & -\epsilon_p & 0 \\ 0 & -\epsilon_p & -\epsilon_s v_1^{'+}(1) + \epsilon_p & -\epsilon_s u_1^{'+}(1) \\ 0 & 0 & -u_1^{'+}(x_1^+) & u_1^{'+}(x_1^+) + \alpha_2^- \end{pmatrix} \begin{pmatrix} \phi_1^- \\ \phi_2^- \\ \phi_0^+ \\ \phi_1^+ \end{pmatrix}$$

$$= \begin{pmatrix} \beta_2^-(u_2^{'-}(x_1^-) + v_2^{'-}(x_1^-)) \\ \epsilon_s(\phi'_{N-1} - \beta_N^-)(u_N^{'-}(0) + v_N^{'-}(0)) + q(1-y) \\ -\epsilon_s(\phi_0^+ - \beta_0^+)(u_1^{'+}(1) + v_1^{'+}(1)) + qy \\ -\beta_1^+(u_1^{'+}(x_1^+) + v_1^{'+}(x_1^+)) \end{pmatrix}. \tag{160}$$

When $N \geq 4$, the matching conditions are $Ax = b$, where

$$\begin{pmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & a_2 & b_2 & c_2 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & a_{N-1} & b_{N-1} & c_{N-1} & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & a_N & b_N & c_N & 0 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 0 & a_{N+1} & b_{N+1} & c_{N+1} & 0 & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & a_{N+2} & b_{N+2} & c_{N+2} & \cdots & 0 \\ 0 & \cdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & a_{2N-1} & b_{2N-1} & c_{2N-1} & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & a_{2N} & b_{2N} \end{pmatrix} \begin{pmatrix} \phi_1^- \\ \phi_2^- \\ \vdots \\ \phi_{N-1}^- \\ \phi_N^- \\ \phi_0^+ \\ \phi_1^+ \\ \vdots \\ \phi_{N-2}^+ \\ \phi_{N-1}^+ \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{N-1} \\ r_N \\ r_{N+1} \\ r_{N+2} \\ \vdots \\ r_{2N-1} \\ r_{2N} \end{pmatrix}, \tag{161}$$

$$\begin{aligned}
&b_1 = u_2^{'-}(x_1^-) + \alpha_1^+, \quad c_1 = -u_2^{'-}(x_1^-), \\
&a_2 = v_2^{'-}(x_2^-), \quad b_2 = u_2^{'-}(x_2^-) - v_3^{'-}(x_2^-), \quad c_2 = -u_3^{'-}(x_2^-), \\
&a_{N-1} = v_{N-1}^{'-}(x_{N-1}^-), \quad b_{N-1} = u_{N-1}^{'-}(x_{N-1}^-) - v_N^{'-}(x_{N-1}^-), \quad c_{N-1} = -u_N^{'-}(x_{N-1}^-), \\
&a_N = \epsilon_s v_N^{'-}(0), \quad b_N = \epsilon_s u_N^{'-}(0) + \epsilon_p, \quad c_N = -\epsilon_p, \\
&a_{N+1} = -\epsilon_p, \quad b_{N+1} = -\epsilon_s v_1^{'+}(1) + \epsilon_p, \quad c_{N+1} = -\epsilon_s u_1^{'+}(1), \\
&a_{N+2} = v_1^{'+}(x_1^+), \quad b_{N+2} = u_1^{'+}(x_1^+) - v_2^{'+}(x_1^+), \quad c_{N+2} = -u_2^{'+}(x_1^+), \\
&a_{2N-1} = v_{N-2}^{'+}(x_{N-2}^+), \quad b_{2N-1} = u_{N-2}^{'+}(x_{N-2}^+) - v_{N-1}^{'+}(x_{N-2}^+), \quad c_{2N-1} = -u_{N-1}^{'+}(x_{N-2}^+), \\
&a_{2N} = -u_1^{'+}(x_1^+), \quad b_{2N} = u_1^{'+}(x_1^+) + \alpha_2^-,
\end{aligned} \tag{162}$$

$$\begin{aligned}
&r_1 = \beta_2^-(u_2^{'-}(x_1^-) + v_2^{'-}(x_1^-)), \\
&r_2 = (\phi_1^- - \beta_2^-)(u_2'^-(x_2^-) + v_2'^-(x_2^-)) - (\phi_2^- - \beta_3^-)(u_3'^-(x_2^-) + v_3'^-(x_2^-)), \\
&r_{N-1} = (\phi_{N-2}^- - \beta_{N-1}^-)(u_{N-1}'^-(x_{N-1}^-) + v_{N-1}'^-(x_{N-1}^-)) - (\phi_{N-1}^- - \beta_N^-)(u_N'^-(x_{N-1}^-) + v_N'^-(x_{N-1}^-)), \\
&r_N = \epsilon_s(\phi_{N-1}' - \beta_N^-)(u_N^{'-}(0) + v_N^{'-}(0)) + q(1-y), \\
&r_{N+1} = -\epsilon_s(\phi_0^+ - \beta_0^+)(u_1^{'+}(1) + v_1^{'+}(1)) + qy, \\
&r_{N+2} = (\phi_0^+ - \beta_1^+)(u_1'^+(x_1^+) + v_1'^+(x_1^+)) - (\phi_1^- - \beta_2^+)(u_2'^+(x_1^+) + v_2'^+(x_1^+)), \\
&r_{2N-1} = (\phi_{N-3}^+ - \beta_{N-2}^+)(u_{N-2}'^+(x_{N-2}^+) + v_{N-2}'^+(x_{N-2}^+)) - (\phi_{N-2}^+ - \beta_{N-1}^+)(u_{N-1}'^+(x_{N-2}^+) + v_{N-1}'^+(x_{N-2}^+)), \\
&r_{2N} = -\beta_{N-1}^+(u_{N-1}^{'+}(x_{N-1}^+) + v_{N-1}^{'+}(x_{N-1}^+)).
\end{aligned} \tag{163}$$

In Figure 8, the numerical results are given after building in Matlab, with input test physical

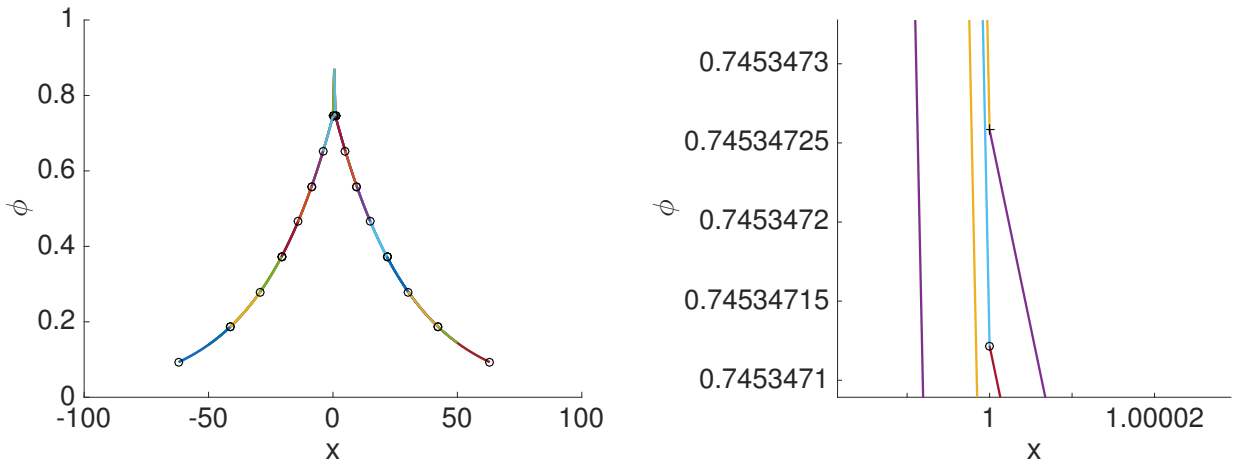

Figure 8: Numerical results: 8-Interval solution (left), magnification around ϕ_0^+ point (right)

constants,

$$\epsilon_s = 20,\ \epsilon_p = 2,\ q = 1,\ \kappa = 0.15,\ y = 0.5, \tag{164}$$

and the error tolerance is set as 1e-12. The left graph in Figure 8 is the magnification around the ϕ_0^+ point, which only left with the exact solution with the plus symbol and the 8-interval solution with a circle symbol. There are two curves on the right of the 8-interval solution are 4-interval solution and 2-intervals solution. The 4-interval solution is closer to the 8-interval solution which means the algorithm converges monotonically. The Figure 9 contains two different rate of magnification around the same point ϕ_0^+. In the left graph of Figure 9, compare to the right graph of Figure 8, the lowest circle comes from the 4-interval solution, with the left two curves 2-interval solution and 1-interval solution similarly. The same as the left graph. Among them, we can discover for each iteration, the new numerical solution converges closer to the exact solution which is about one-forth distance of the previous numerical solution. The graphs show that the rate of convergence is quite fast.

The error analysis matches our expectation of $O(N^{-2})$ in Table 4. Besides, the third column in Table 4 also shows that the converged constant calculated by max error times N^2 is a quite small number under 1e-6 scale. Later, we will change different physical constants and see how the max error changes.

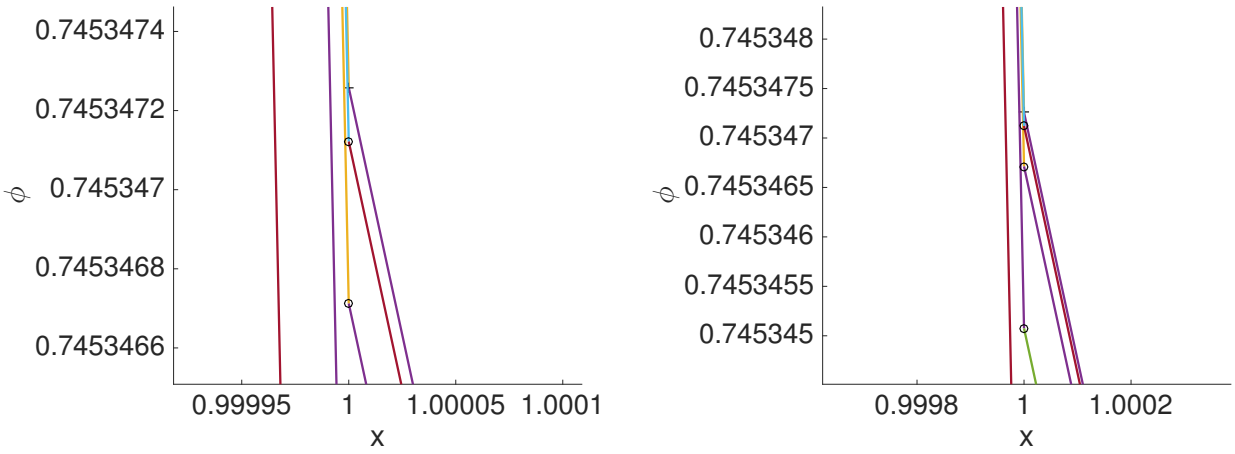

Figure 9: Numerical results of 8-Interval solution magnification around ϕ_0^+ point

N	Max Error	Max Error· N^2
1	0.0000089288	0.0000089288
2	0.0000021887	0.0000087550
4	0.0000005461	0.0000087380
8	0.0000001364	0.0000087349
16	0.0000000341	0.0000087342
32	0.0000000085	0.0000087339
64	0.0000000021	0.0000087334

Table 4: Numerical results: errors analysis for 2^k number of intervals, k = 0 : 6

5 Conclusion

In the report, we first explained how to use the Green's function method to solve two-point boundary differential equations and did the numerical evaluation and error analysis. Then, we introduce the definition and properties of pKa. At last, we focus on solving linear and nonlinear PBE. A numerical technique called quasilinearization is applied to the solve the nonlinear PBE which approximates the nonlinear term by piecewise interpolated lines. The numerical result showing that the scheme converges at the rate of $O(N^{-2})$, where N is the number of the intervals in the approximation. Currently, we are applying the quasilinearization technique to the 1D solvent/protein/solvent model and do the corresponding analysis. In the future, we will see how to apply to the 2D/3D models and compute pKa value.

References

[1] Greenberg, Michael D., *Applications of Green's Functions in Science and Engineering*, 3-20, (2015).

[2] Xiao L., Q. Cai, X. Ye, J. Wang and R. Luo, *Electrostatic forces in the Poisson-Boltzmann systems*, J. Chem. Physics, **139**, 094106 (2013).

[3] Geng W. and Krasny R., *A treecode-accelerated boundary integral Poisson-Boltzmann solver for electrostatics of solvated biomolecules.*, J. Comput. Physics, **247**, 62-79 (2013).

[4] Andrew Lucas website. `http://people.fas.harvard.edu/~lucas/problems.html`

[5] R. C. Y. Chin and R. Krasny, *A hybrid asymptotic-finite method for stiff two-point boundary value problem*, SIAM J. Sci. Stat. Comput., **4**, 229-243 (1983).