

Computing Protein pKas Using the TABI Poisson–Boltzmann Solver

Jiahui Chen^{*,¶}, Jingzhen Hu^{†,||}, Yongjia Xu^{‡,**,††}, Robert Krasny^{§,††} and Weihua Geng^{‡,††}^{*}Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA[†]Department of Mathematics, Duke University, Durham, NC 27710, USA[‡]Department of Mathematics, Southern Methodist University, Dallas, TX 75275, USA[§]Department of Mathematics, University of Michigan, Ann Arbor, MI 48109, USA[¶]chenj159@msu.edu^{||}jingzhen.hu@duke.edu^{**}yongjiax@smu.edu^{††}krasny@umich.edu^{‡‡}Corresponding author. E-mail: wgeng@smu.edu

ABSTRACT: A common approach to computing protein pKas uses a continuum dielectric model in which the protein is a low dielectric medium with embedded atomic point charges, the solvent is a high dielectric medium with a Boltzmann distribution of ionic charges, and the pKa is related to the electrostatic free energy which is obtained by solving the Poisson–Boltzmann equation. Starting from the model pKa for a titrating residue, the method obtains the intrinsic pKa and then computes the protonation probability for a given pH including site–site interactions. This approach assumes that acid dissociation does not affect protein conformation aside from adding or deleting charges at titratable sites. In this work, we demonstrate our treecode-accelerated boundary integral (TABI) solver for the relevant electrostatic calculations. The pKa computing procedure is enclosed in a convenient Python wrapper which is publicly available at the corresponding author's website. Predicted results are compared with experimental pKas for several proteins. Among ongoing efforts to improve protein pKa calculations, the advantage of TABI is that it reduces the numerical errors in the electrostatic calculations so that attention can be focused on modeling assumptions.

KEYWORDS: Protein pKa; acid dissociation constant; continuum dielectric model; Poisson–Boltzmann equation; boundary integral equation; treecode.

1. INTRODUCTION

Proton transfer between the ionizable amino acids of a solvated protein and the surrounding aqueous solvent has an important effect on pH-dependent processes such as enzymatic activity¹ and protein–ligand binding.² In the ideal case of an acid with a single ionizable group, the pKa can be measured by simple acid–base titration. The acid dissociation reaction, $HA \rightleftharpoons A^- + H^+$, has equilibrium constant defined by

$$K_a = \frac{[A^-][H^+]}{[HA]}, \quad (1)$$

where $[HA]$, $[A^-]$, $[H^+]$ are the equilibrium concentrations of the protonated acid, its deprotonated conjugate,

and free protons. The Henderson–Hasselbalch equation,

$$pH = pK_a + \log_{10} \frac{[A^-]}{[HA]}, \quad (2)$$

relates the solvent $pH = -\log_{10}[H^+]$ with the acid $pK_a = -\log_{10}K_a$ and shows that $pH = pK_a$ when $[HA] = [A^-]$, i.e. when the acid is equally likely to be protonated or deprotonated.

Proteins however have multiple ionizable groups that titrate at different rates depending on the solution pH, and measuring the pKas of the various sites is more difficult. One approach to determining protein pKas uses chemical shifts measured experimentally by NMR

This is an Open Access article. It is distributed under the terms of the Creative Commons Attribution 4.0 (CC-BY) License. Further distribution of this work is permitted, provided the original work is properly cited.

Received: 6 August 2020

Accepted: 6 November 2020

Published: 10 December 2020

OPEN ACCESS

DOI: 10.1142/S2737416520420065

J. Comput. Biophys. Chem. 2021, 20 (2), 175–187

spectroscopy,³ Alternatively, protein pKas can be predicted theoretically,⁴ and methods reported in the literature employ continuum dielectric models,^{5–7} Monte Carlo methods,⁸ molecular dynamics and QM/MM simulations,^{9,10} and empirical relationships.^{11,12}

This work focuses on a continuum dielectric model in which the protein pKas are given by electrostatic free energy shifts, where the solvation free energy is obtained by solving the Poisson–Boltzmann (PB) equation.^{5,13,14} The protein is represented by a low dielectric cavity with embedded atomic partial charges, and the surrounding aqueous solvent is a high dielectric medium with dissolved ions.^{15,16} The dielectric interface is most commonly assumed to be the molecular surface (or solvent-excluded surface).¹⁷ The model assumes that protonation/deprotonation has negligible effect on the protein conformation, and the electrostatic free energy shifts arise entirely from changes in the protonation state of the ionizable groups. Under this assumption, the same protein structure is used for all protonation states, and the pKa computation requires solving the PB equation many times with different charge distributions to obtain the electrostatic free energy shifts. This calls for an accurate and efficient numerical method to handle the challenges of PB simulations which arise due to the irregular geometry of the molecular surface, the jump in dielectric constant across the surface, the singularity of the atomic partial charges representing the protein, and the infinite domain on which the problem is posed.

Among several numerical PB solvers developed in the literature, the treecode-accelerated boundary integral (TABI) method is chosen for the pKa computations in this work. The boundary integral formulation used in TABI analytically satisfies the dielectric jump conditions and accounts for the singular charges and infinite domain, while the irregular geometry of the molecular surface is resolved by triangulation. Compared to alternative grid-based PB solvers, the TABI solver is particularly useful in treating the complex charge distributions necessary in pKa calculations. Our numerical results show that TABI can provide accurate electrostatic free energies and protein pKas even with a relatively coarse triangulation. A Python wrapper was written to pipeline the entire pKa computing procedure using TABI and is publicly available.

The rest of this article is organized as follows. Section 2 presents the continuum dielectric model based on the PB equation. Section 3 presents the boundary integral form of the PB equation and the TABI solver. Section 4 reviews how protein pKas are computed from statistical mechanics. Section 5 describes the pKa

calculating process and its implementation in a Python wrapper. Section 6 reports results in computing pKas for two sample proteins. The article ends with a conclusion section summarizing the main points.

2. CONTINUUM DIELECTRIC MODEL

Figure 1 illustrates the model showing the protein domain Ω_p , solvent domain Ω_s , and the interface Γ between Ω_p and Ω_s , which in this work is taken to be the molecular surface (or solvent excluded surface).¹⁸ Partial charges q_k are located at atoms $\mathbf{y}_k \in \Omega_p$ representing the protein for $k = 1 : N_c$, while a Boltzmann charge distribution in Ω_s represents the dissolved ions. Applying Gauss's law to the charge distributions in Ω_p, Ω_s , and taking the limit of weak ionic strength leads to the linear PB equation for the electrostatic potential,

$$-\nabla \cdot (\epsilon(\mathbf{x}) \nabla \phi(\mathbf{x})) + \bar{\kappa}^2(\mathbf{x}) \phi(\mathbf{x}) = 4\pi \sum_{k=1}^{N_c} q_k \delta(\mathbf{x} - \mathbf{y}_k), \quad \mathbf{x} \in \Omega_p, \Omega_s. \quad (3)$$

The potential and electric displacement are continuous across the interface,

$$\phi_p(\mathbf{x}) = \phi_s(\mathbf{x}), \quad \epsilon_p \frac{\partial \phi_p(\mathbf{x})}{\partial n} = \epsilon_s \frac{\partial \phi_s(\mathbf{x})}{\partial n}, \quad \mathbf{x} \in \Gamma, \quad (4)$$

where the subscripts indicate limiting values from either domain, and the potential satisfies the far-field boundary condition, $\phi(\mathbf{x}) \rightarrow 0$ as $|\mathbf{x}| \rightarrow \infty$. The dielectric constant and screening parameter are taken to

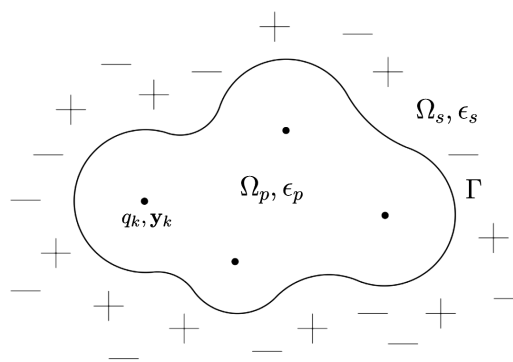


Fig. 1. Dielectric continuum model of solvated protein, protein domain Ω_p with dielectric constant ϵ_p and partial charges q_k at atom locations $\mathbf{y}_k, k = 1 : N_c$, solvent domain Ω_s with dielectric constant ϵ_s and dissolved ions (+/-), molecular surface Γ .

be piecewise constant,

$$\epsilon(\mathbf{x}) = \begin{cases} \epsilon_p, & \mathbf{x} \in \Omega_p \\ \epsilon_s, & \mathbf{x} \in \Omega_s \end{cases}, \quad \bar{\kappa}^2(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \Omega_p \\ \epsilon_s \kappa^2, & \mathbf{x} \in \Omega_s \end{cases}, \quad (5)$$

where κ is the inverse Debye length determined by the ion concentration I in the solvent (see Refs. 19 and 20 for details on the definition and units of these parameters). We consider several values for the protein dielectric constant ϵ_p , with solvent dielectric constant $\epsilon_s = 80$ and inverse Debye length $\kappa = 0.1257 \text{ \AA}^{-1}$ corresponding to ion concentration $I = 0.15 \text{ M}$.

The PB equation (3) has analytic solutions only for simple interface geometries such as a plane or sphere, and for general molecular surfaces, it is solved numerically using methods described in several comprehensive reviews.^{21,22} Numerical solution of the PB equation faces several challenges, in particular the protein molecular surface is geometrically complex, the dielectric constant is discontinuous across the molecular surface, the protein is represented by singular atomic partial charges, and the domain is unbounded. Numerical methods fall into two main categories: *grid-based methods* discretize the equation directly on a volumetric grid using finite-difference or finite-element approximations,^{23–30} while *boundary element methods* solve an equivalent integral equation on the triangulated molecular surface.^{31–38} Boundary element methods effectively address the challenges mentioned above, but they result in dense linear systems which can be expensive to solve by iterative methods such as GMRES.³⁹ In this work, we employ a treecode to compute the dense matrix–vector products required in GMRES iteration, thereby reducing the operation count from $O(N^2)$ to $O(N \log N)$, where N is the number of faces in the triangulation of the molecular surface. The resulting TABI solver³⁵ is described in the next section.

3. TABI SOLVER

This section describes the boundary integral form of the PB equation,³³ then the discretization of the integral equations, then the treecode algorithm for fast matrix–vector product and finally the choice of the MSMS density parameter for triangulating the molecular surface.

3.1. Boundary integral form of PB equation

Applying Green's second identity to Eq. (3), the electrostatic potential in each domain can be expressed in

terms of single-layer and double-layer potentials on the molecular surface³³

$$\phi(\mathbf{x}) = \int_{\Gamma} \left[G_0(\mathbf{x}, \mathbf{y}) \frac{\partial \phi(\mathbf{y})}{\partial n} - \frac{\partial G_0(\mathbf{x}, \mathbf{y})}{\partial n} \phi(\mathbf{y}) \right] dS_{\mathbf{y}} + \sum_{k=1}^{N_c} q_k G_0(\mathbf{x}, \mathbf{y}_k), \quad \mathbf{x} \in \Omega_p, \quad (6a)$$

$$\phi(\mathbf{x}) = \int_{\Gamma} \left[-G_{\kappa}(\mathbf{x}, \mathbf{y}) \frac{\partial \phi(\mathbf{y})}{\partial n} + \frac{\partial G_{\kappa}(\mathbf{x}, \mathbf{y})}{\partial n} \phi(\mathbf{y}) \right] dS_{\mathbf{y}}, \quad \mathbf{x} \in \Omega_s, \quad (6b)$$

where the Coulomb potential and screened Coulomb potential are

$$G_0(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi|\mathbf{x} - \mathbf{y}|}, \quad G_{\kappa}(\mathbf{x}, \mathbf{y}) = \frac{e^{-\kappa|\mathbf{x} - \mathbf{y}|}}{4\pi|\mathbf{x} - \mathbf{y}|}. \quad (7)$$

The singular charges in the protein domain are represented by the Coulomb potential source term in Eq. (6a), and the far-field boundary condition, $\phi(\mathbf{x}) \rightarrow 0$ as $|\mathbf{x}| \rightarrow \infty$, is analytically satisfied.

Enforcing the interface conditions in Eq. (4) yields a set of boundary integral equations relating the surface potential ϕ_1 and its normal derivative $\partial \phi_1 / \partial n$ on Γ ,

$$\frac{1}{2}(1 + \epsilon)\phi_1(\mathbf{x}) = \int_{\Gamma} \left[K_1(\mathbf{x}, \mathbf{y}) \frac{\partial \phi_1(\mathbf{y})}{\partial n} + K_2(\mathbf{x}, \mathbf{y}) \phi_1(\mathbf{y}) \right] dS_{\mathbf{y}} + S_1(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \quad (8a)$$

$$\frac{1}{2}(1 + \epsilon^{-1}) \frac{\partial \phi_1(\mathbf{x})}{\partial n} = \int_{\Gamma} \left[K_3(\mathbf{x}, \mathbf{y}) \frac{\partial \phi_1(\mathbf{y})}{\partial n} + K_4(\mathbf{x}, \mathbf{y}) \phi_1(\mathbf{y}) \right] dS_{\mathbf{y}} + S_2(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \quad (8b)$$

where $\epsilon = \epsilon_2 / \epsilon_1$ is the ratio of dielectric constants. The kernels $K_{1,2,3,4}$ and source terms $S_{1,2}$ are linear combinations of G_0 , G_{κ} and their first- and second-order normal derivatives.^{33,35} Note that Eqs. (8a) and (8b) are coupled integral equations of the second kind, and together with the properties of the kernels, this ensures that they are well-conditioned. The electrostatic free energy of the solvated protein is

$$\Delta G = \frac{1}{2} \sum_{\substack{j,k=1 \\ j \neq k}}^{N_c} \frac{q_j q_k}{|\mathbf{y}_j - \mathbf{y}_k|} + \frac{1}{2} \sum_{k=1}^{N_c} q_k \phi_{\text{reac}}(\mathbf{y}_k), \quad (9)$$

where the first term on the right is the Coulomb energy and the second term is the electrostatic solvation energy, and the reaction potential at the k th charge site is

$$\begin{aligned} \phi_{\text{reac}}(\mathbf{y}_k) \\ = \int_{\Gamma} \left[K_1(\mathbf{y}_k, \mathbf{y}) \frac{\partial \phi_1(\mathbf{y})}{\partial n} + K_2(\mathbf{y}_k, \mathbf{y}) \phi_1(\mathbf{y}) \right] dS_{\mathbf{y}}. \end{aligned} \quad (10)$$

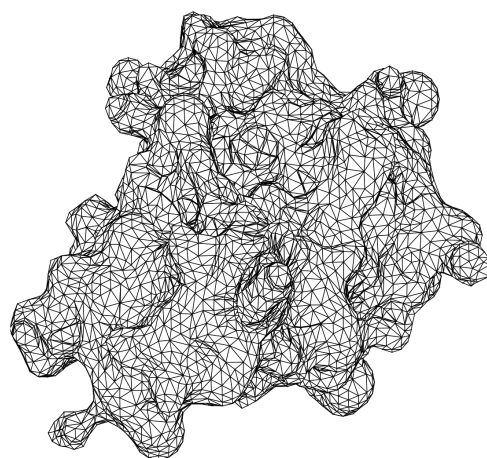
3.2. Discretization of boundary integral equations

The molecular surface Γ is triangulated using MSMS,¹⁸ where the density parameter d is the number of vertices per $\text{\AA}^2$ of surface area. Figure 2 shows two examples with a relatively coarse density $d = 2$, (a) BPTI, (b) OMTKY3. Letting $\mathbf{x}_i$, $i = 1 : N$ denote the triangle centroids, the integrals in Eqs. (8a) and (8b) are discretized by a centroid collocation boundary element method,

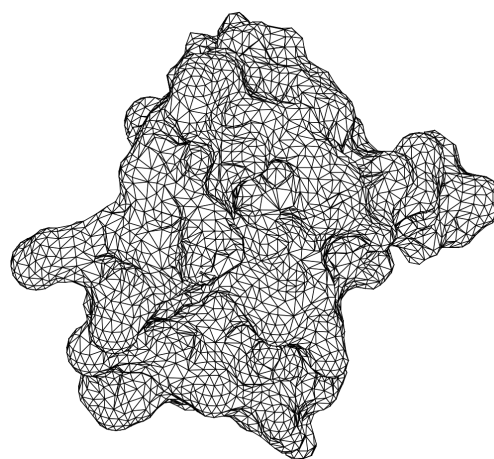
$$\begin{aligned} \frac{1}{2} (1 + \varepsilon) \phi_1(\mathbf{x}_i) \\ = \sum_{\substack{j=1 \\ j \neq i}}^N \left[K_1(\mathbf{x}_i, \mathbf{x}_j) \frac{\partial \phi_1(\mathbf{x}_j)}{\partial n} + K_2(\mathbf{x}_i, \mathbf{x}_j) \phi_1(\mathbf{x}_j) \right] A_j \\ + S_1(\mathbf{x}_i), \end{aligned} \quad (11a)$$

$$\begin{aligned} \frac{1}{2} (1 + \varepsilon^{-1}) \frac{\partial \phi_1(\mathbf{x}_i)}{\partial n} \\ = \sum_{\substack{j=1 \\ j \neq i}}^N \left[K_3(\mathbf{x}_i, \mathbf{x}_j) \frac{\partial \phi_1(\mathbf{x}_j)}{\partial n} + K_4(\mathbf{x}_i, \mathbf{x}_j) \phi_1(\mathbf{x}_j) \right] A_j \\ + S_2(\mathbf{x}_i), \end{aligned} \quad (11b)$$

where A_j is the triangle area, and the $j = i$ term in the sums is omitted to avoid the kernel singularity. Equation (10) is discretized in a similar way to obtain the reaction potential. Equations (11a) and (11b) form a $2N \times 2N$ linear system for the values of the surface potential $\phi_1(\mathbf{x}_i)$ and its normal derivative $\partial \phi_1(\mathbf{x}_i)/\partial n$ at the triangle centroids. The system is solved by GMRES iteration which requires a matrix–vector product at each step.³⁹ Since the matrix is dense, computing the product by direct summation requires $O(N^2)$ operations, which is prohibitively expensive when N is large. The next subsection describes the treecode algorithm used to reduce the cost of the matrix–vector product.



(a)



(b)

Fig. 2. Triangulation of protein molecular surfaces using MSMS¹⁸ with density $d = 2$: (a) bovine pancreatic trypsin inhibitor (BPTI, PDB ID 4pti) and (b) turkey ovomucoid third domain (OMTKY3, PDB ID 2ovo).

3.3. Treecode acceleration of matrix–vector product

We summarize the treecode algorithm and refer to previous work for more detail.^{40,41} The matrix–vector product arising from Eqs. (11a) and (11b) consists of terms in the form of charged particle interactions,

$$V_i = \sum_{\substack{j=1 \\ j \neq i}}^N K(\mathbf{x}_i, \mathbf{x}_j) q_j, \quad i = 1, \dots, N, \quad (12)$$

where V_i is the potential at a target particle $\mathbf{x}_i$, K is one of the kernels, and q_j is a charge associated with the source particle $\mathbf{x}_j$. To evaluate the potentials V_i rapidly, the particles $\mathbf{x}_i$ are divided into a hierarchy of clusters having a tree structure as depicted in Fig. 3(a). Then V_i

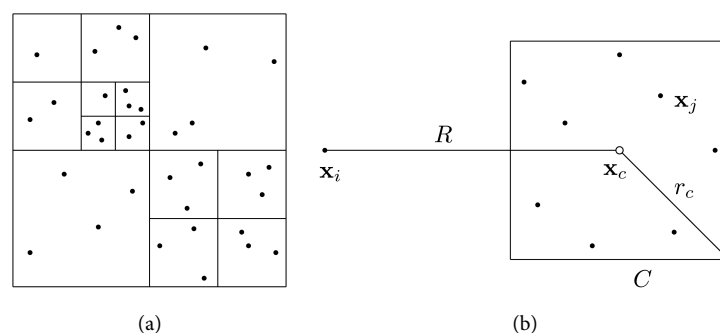


Fig. 3. Treecode schematic in 2D: (a) tree structure with four levels of particle clusters; (b) interaction between particle $\mathbf{x}_i$ and cluster $C = \{\mathbf{x}_j\}$, cluster center $\mathbf{x}_c$, cluster radius r_c , particle-cluster distance $R = |\mathbf{x}_i - \mathbf{x}_c|$.

is evaluated as a sum of particle–cluster interactions,

$$V_i \approx \sum_{C \in N_i} \sum_{\mathbf{x}_j \in C} K(\mathbf{x}_i, \mathbf{x}_j) q_j + \sum_{C \in F_i} \sum_{|\mathbf{k}|=0}^p \times a_{\mathbf{k}}(\mathbf{x}_i, \mathbf{x}_c) m_{\mathbf{k}}(C), \quad (13)$$

where C denotes a cluster, and N_i, F_i denote the near-field and far-field clusters of particle $\mathbf{x}_i$. Figure 3(b) depicts a single particle–cluster interaction. The first term on the right-hand side of Eq. (13) is a direct sum for particles $\mathbf{x}_j$ near $\mathbf{x}_i$, while the second term is a p th order Cartesian Taylor approximation about the cluster center $\mathbf{x}_c$, for clusters that are well-separated from $\mathbf{x}_i$, where $a_{\mathbf{k}}(\mathbf{x}_i, \mathbf{x}_c)$ are the Taylor coefficients and $m_{\mathbf{k}}(C)$ are the cluster moments.⁴¹

A particle $\mathbf{x}_i$ and cluster C are *well-separated* if the multipole acceptance criterion (MAC) is satisfied, $r_c/R \leq \theta$, where r_c is the cluster radius, $R = |\mathbf{x}_i - \mathbf{x}_c|$ is the particle–cluster distance and θ is a user-specified

parameter.⁴⁰ The treecode cycles through the particles $\mathbf{x}_i$ and interacts with the clusters in the tree; if the MAC is satisfied, then the Taylor approximation is used; otherwise, the child clusters are checked, unless the cluster is a leaf at which point direct summation is used. The accuracy of the treecode is controlled by the Taylor approximation order p and MAC parameter θ . Using the treecode, the operation count for the matrix–vector product is reduced to $O(N \log N)$; the factor N is the number of particles $\mathbf{x}_i$, and the factor $\log N$ is the number of levels in the tree. This completes the description of the TABI solver.³⁵

3.4. Choice of MSMS density

The MSMS density d is critical in determining the accuracy of the TABI solver. To demonstrate this, we computed the electrostatic free energy ΔG for two proteins, bovine pancreatic trypsin inhibitor

Table 1. Electrostatic free energy ΔG in Eq. (9) computed by TABI solver, (a) protein BPTI, (b) protein OMTKY3, protein dielectric constant $\epsilon_p = 4, 20$, MSMS density $d = 5, 10, 20, 40$ vertices/Å², the number displayed for $d = 5, 10, 20$ is the difference in ΔG from the $d = 40$ value in units of kcal/mol, relative error (%) is computed using $d = 40$ value as reference, run time in seconds.

d	ΔG (kcal/mol)		Relative error (%)		Run time (s)	
	$\epsilon_p = 4$	$\epsilon_p = 20$	$\epsilon_p = 4$	$\epsilon_p = 20$	$\epsilon_p = 4$	$\epsilon_p = 20$
(a) BPTI						
5	−17.3	−2.6	0.29	0.22	47.8	43.5
10	−6.0	−0.9	0.10	0.08	106.4	91.3
20	−1.8	−0.3	0.03	0.02	277.9	205.5
40	−5888.5	−1162.9	—	—	816.9	461.5
(b) OMTKY3						
5	−11.2	−1.6	0.26	0.19	56.2	83.7
10	−3.9	−0.6	0.09	0.07	100.2	79.2
20	−1.2	−0.2	0.03	0.02	194.4	169.4
40	−4326.8	−852.6	—	—	468.2	381.8

(BPTI, PDB ID: 4pti)⁴² and turkey ovomucoid third domain (OMTKY3, PDB ID: 2ovo).⁴³ The treecode used MAC parameter $\theta = 0.8$ and order $p = 3$, which ensures that the treecode approximation error is less than the discretization error in the boundary element method. We use PDB2PQR⁴⁴ to assign partial charges to the PDB protein structures based on the PARSE force field.⁴⁵ Table 1 presents results for MSMS density $d = 5, 10, 20, 40$ and protein dielectric constant $\epsilon_p = 4, 20$; the number displayed for $d = 5, 10, 20$ is the difference in ΔG from the $d = 40$ value in units of kcal/mol, followed by the relative percent error using the $d = 40$ value as the reference. The last column in Table 1 shows the run time in seconds using a MacBook Pro with 2.2 GHz i7 Intel Core and 16 GB 1600 MHz DDR3 memory; these results used block diagonal preconditioning to reduce the number of GMRES iterations.⁴⁶ The ΔG values vary significantly with the protein dielectric constant ϵ_p , but for each choice of ϵ_p , the computed ΔG converges to the reference value as the MSMS density d increases. With density $d = 10$, the relative error is 0.1% or less, and the run time is less than 2 min; this density value is used in the pKa computations below.

4. PROTEIN pKa FROM FREE ENERGY SHIFTS

The equilibrium constant of the acid dissociation reaction is related to the Gibbs free energy,

$$\Delta G = -RT \ln K_a = RT \ln 10 \cdot \text{pKa}, \quad (14)$$

where $R = 8.31 \text{ J/(mol K)}$ is the gas constant and $T = 298 \text{ K}$ is the temperature in this work, and this enables protein pKas to be computed from free energy shifts due to protonation/deprotonation.⁵ Figure 4 shows the thermodynamic cycle for acid dissociation of a titrating residue in a protein, where AH and A^- stand

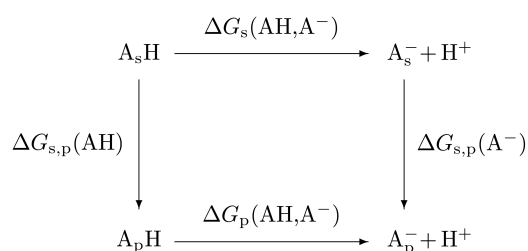


Fig. 4. Thermodynamic cycle for acid dissociation of a titrating residue in a protein, AH/A^- = protonated/deprotonated residue, subscript s/p = solvent/protein environment.

for the protonated and deprotonated states of the residue, respectively, and subscripts s and p stand for the solvent and protein environments, respectively.⁴⁷ The cycle expresses the free energy shifts on transferring the residue from the solvent environment to the protein environment, in its protonated and deprotonated states.

The top arrow in the cycle corresponds to deprotonating the isolated amino acid in the solvent environment and yields the model pKa,

$$\text{pKa}^0 = \frac{1}{RT \ln 10} \Delta G_s(AH, A^-), \quad (15)$$

which can be determined experimentally. Table 2 records the model pKa values for titrating amino acids.⁴⁸ The bottom arrow in the cycle corresponds to deprotonating the amino acid in the protein environment, which can be expressed as

$$\begin{aligned} \Delta G_p(AH, A^-) &= \Delta G_s(AH, A^-) + \Delta G_{s,p}(A^-) \\ &\quad - \Delta G_{s,p}(AH), \end{aligned} \quad (16)$$

where the first term on the right is the free energy for deprotonating the amino acid in the solvent environment, and the last two terms give the free energy shift for moving the indicated group from the solvent to the protein environment. It is assumed here that the protein conformation remains fixed during acid dissociation, and that nonpolar effects are negligible, and hence, the ΔG values in this work refer to the electrostatic free energy in Eq. (9). Starting from the model pKa for a given ionizable group on an amino acid side chain, the procedure computes the intrinsic pKa of the site and then takes site-site interactions into account to determine the site pKa by computational titration.^{5,14,49}

5. PROCEDURE FOR COMPUTING PROTEIN pKas

The computational procedure is composed of four steps described in the following.

Step 1. Prepare PQR files

This step prepares the PQR files for the protein in various protonation states. The protein structure is obtained from the Protein Data Bank (www.pdb.org), and charges are assigned using PDB2PQR⁴⁴ with protonation states specified using the residue IDs in Table 2. For a protein with N_t titrating sites, we need a total of $1 + 2N_t + \frac{1}{2}(N_t^2 - N_t)$ PQR files with charge distributions specified as follows: (a) one file with all titrating sites deprotonated, keeping the background

charges of nontitrating sites on, (b) $2N_t$ files having all titrating sites but one deprotonated, once with background charges on and once with background charges off, (c) $\frac{1}{2}(N_t^2 - N_t)$ files with pairs of titrating sites protonated and all other titrating sites deprotonated, and with background charges off.

Step 2. Call TABI solver for electrostatics

This step calls the TABI solver to compute the electrostatic free energy ΔG in Eq. (9) for each protonation state represented by a PQR file, where the structure and charge distribution in the file gives the source term on the right side of the PB equation in Eq. (3). The TABI solver is called with input physical parameters (dielectric constants, ion concentration) and numerical parameters (MSMS density, treecode MAC, order of Taylor approximation). After completing this step, ΔG is available for each protonation state, and these are used to calculate the intrinsic pKas in the next step.

Step 3. Compute intrinsic pKas

The intrinsic pKa for the i th titrating site of the protein accounts for single-site effects, i.e. when the i th site is protonated, the other titrating sites are deprotonated, and the background charges are on; the expression is

$$\text{pKa}^{\text{int}}(i) = \text{pKa}^0(i) + \frac{1}{RT \ln 10} (\Delta G_p(AH, A^-) - \Delta G_s(AH, A^-)). \quad (17)$$

The first term on the right, $\text{pKa}^0(i)$, is the model pKa for the residue at the i th titrating site as determined experimentally and recorded in Table 2.⁴⁸ The next two terms are electrostatic free energy shifts; $\Delta G_p(AH, A^-)$ is the shift between the protein with only the i th titrating site protonated and the protein with all titrating sites deprotonated, while $\Delta G_s(AH, A^-)$ is the same shift for the residue in the solvent environment. These shifts are computed using the ΔG values obtained in Step 2; note that TABI uses units of

kcal/mol for free energy, so the computed ΔG values are scaled by $RT \ln 10$ to obtain $\text{pKa}^{\text{int}}(i)$.

Step 4. Titration with site-site interactions

The pKa of a titrating site is the pH at which the site is equally likely to be protonated or deprotonated. In a protein with N_t titrating sites, let $\theta \in \{0, 1\}^{N_t}$ define a protonation state, where θ_i is the i th entry of θ , and $\theta_i = 1$ means that the i th site is protonated, while $\theta_i = 0$ means it is deprotonated. The pKa of the i th site is the pH satisfying $\langle \theta_i, \text{pH} \rangle = 0.5$, where the probability that the i th site is protonated at a particular pH is given by the thermodynamic average,^{5,13,14}

$$\langle \theta_i, \text{pH} \rangle = \frac{\sum_{\theta} \theta_i \exp(-\Delta G(\theta, \text{pH})/RT)}{\sum_{\theta} \exp(-\Delta G(\theta, \text{pH})/RT)}. \quad (18)$$

In this expression, the pH-dependent protonation state energy is defined by

$$\Delta G(\theta, \text{pH}) = RT \ln 10 \sum_i \theta_i (\text{pH} - \text{pKa}^{\text{int}}(i)) + \frac{1}{2} \sum_{i \neq j} \theta_i \theta_j \Delta G_{ij}, \quad (19)$$

where $\text{pKa}^{\text{int}}(i)$ is the intrinsic pKa of the i th site computed in Step 3. Equation (19) also requires the site-site interaction energy defined by

$$\begin{aligned} \Delta G_{ij} &= \frac{1}{2} (t_i^T W t_j + t_j^T W t_i) \\ &= \frac{1}{2} (t_i + t_j)^T W (t_i + t_j) \\ &\quad - \frac{1}{2} t_i^T W t_i - \frac{1}{2} t_j^T W t_j, \end{aligned} \quad (20)$$

where t_i is a vector of partial charges where the charges of the i th titrating site are on, and the charges of the other titrating sites and background charges are off, while W is the mapping from partial charges to potential values at the charge sites as computed by the TABI solver. The first equality in Eq. (20) expresses

Table 2. Titrating amino acids, residue ID, residue type, model pKa (pKa^0) determined experimentally,⁴⁸ residue ID for protonated/deprotonated states with total charge in parentheses.

Residue ID	Residue type	pKa^0	Protonated	Deprotonated
ASP	Acidic	4.0	ASH (0)	ASP (−1)
GLU	Acidic	4.4	GLH (0)	GLU (−1)
CYS	Polar uncharged	9.5	CYX (0)	CYM (−1)
TYR	Polar uncharged	9.6	TYR (0)	TYM (−1)
HIS	Basic	6.3	HIP (+1)	HIE (0)
LYS	Basic	10.4	LYS (+1)	LYN (0)

ΔG_{ij} as the symmetric interaction of site i and site j , while the second equality in Eq. (20) reduces the cost of computing the ΔG_{ij} values from $O(N_t^2)$ to $O(N_t)$.

Since there are 2^{N_t} protonation states, computing the thermodynamic average in Eq. (18) explicitly is only feasible for relatively small proteins. The proteins considered in this work (BPTI, OMTKY3) were chosen for two reasons: (1) they have few enough titrating sites so that the thermodynamic average can be computed explicitly, enabling us to avoid numerical errors introduced by approximating $\langle \theta_i, \text{pH} \rangle$, and (2) experimental pKas are available for most of the titrating sites.¹² Future work will consider combining the TABI solver for electrostatics with more efficient procedures for computing the thermodynamic average such as the reduced-site approximation,^{5,50} Monte Carlo methods,^{13,14,51} and clustering algorithms.⁵²

5.1. Examples of titration curves

For each titrating site, the protonation probability $\langle \theta_i, \text{pH} \rangle$ is sampled from $\text{pH} = 1$ to $\text{pH} = 14$ with step size of 0.2, and cubic spline interpolation is used to find the pH at probability 1/2 giving the predicted site pKa. In this way, each active site has a titration curve and Fig. 5 shows two examples for which the titration curve has the conventional sigmoidal shape. Figure 5(a) is for the basic residue LYS15 on protein BPTI at $\epsilon_p = 20$, which predicts $\text{pKa} = 10.47$ versus the experimental value 10.4, while Fig. 5(b) is for the acidic residue GLU43 on protein OMTKY3 at $\epsilon_p = 8$, which predicts $\text{pKa} = 4.79$ versus the experimental value 4.8. The results indicate that our computational procedure is capable of achieving good agreement with experiment, although already there is a suggestion that the

choice of protein dielectric constant ϵ_p plays an important role.

5.2. Python wrapper

A convenient Python wrapper was written to pipeline the entire pKa calculating procedure and is publicly available for download on the corresponding author's website (faculty.smu.edu/wgeng/research/pka-tabi.html) at Southern Methodist University; also available there is the TABI source code for electrostatics and multi-platform binary versions of MSMS¹⁸ for molecular surface triangulation. The user specifies parameters for the PB equation (dielectric constants, ion concentration) and the TABI solver (MSMS density, treecode MAC, order of Taylor approximation) in the user data file. On a computer with Python and Fortran compilers installed, and the protein specified by its four-digit PDB ID, the user runs the wrapper by typing the following command:

```
python wrapper_pka.py PDBID.
```

The wrapper downloads the protein structure from the Protein Data Bank, identifies the titrating sites, calls the TABI solver for electrostatics, computes the intrinsic pKa and protonation probability of each site including site-site interactions as described above, and returns the predicted pKas.

6. NUMERICAL RESULTS

Table 3 presents results for and OMTKY3 with two values of the protein dielectric constant, $\epsilon_p = 4, 20$. Successive columns give the residue ID, model pKa (pKa^0), experimental pKa (pKa^{exp}), pKa shift with

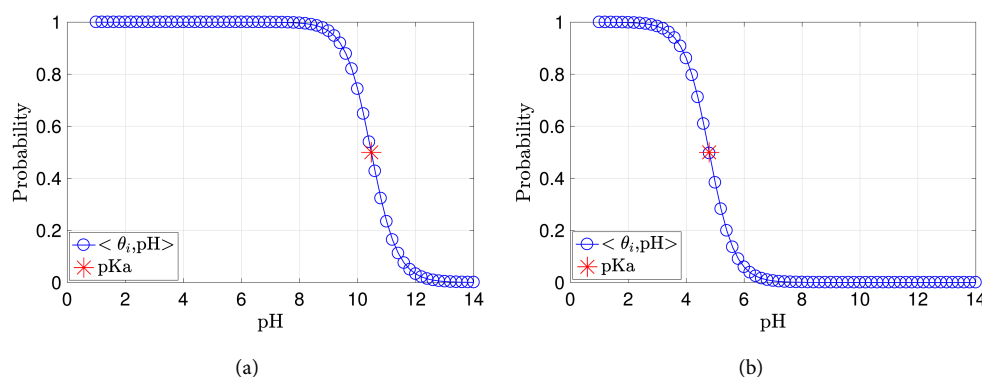


Fig. 5. Computed titration curves, protonation probability of ionizable residue is plotted versus pH, blue circles ($\langle \theta_i, \text{pH} \rangle$), red stars (predicted pKa): (a) LYS15 on protein BPTI, predicted pKa = 10.47 ($\epsilon_p = 20$), experimental pKa = 10.4; (b) GLU43 on protein OMTKY3, predicted pKa = 4.79 ($\epsilon_p = 8$), experimental pKa = 4.8.

respect to experiment ($\Delta pK_a^{0,exp}$), intrinsic pKa (pK_a^{int}), and predicted pKa (pK_a^{pre}); both proteins have 12 titrating residues for which experimental pKas are available.¹² The bold values in each row are the closest to experiment, and the RMSD is with respect to the experimental pKa.

For BPTI, the model pKa and intrinsic pKa are each closest to experiment for five sites, while the predicted pKa is closest for two sites. The RMSD for the model pKa is 0.63 pK units. At the lower dielectric constant $\epsilon_p = 4$, the RMSD for the intrinsic pKa and predicted pKa is more than 2 pK units, while at the higher value $\epsilon_p = 20$ they are essentially the same as the model pKa RMSD. For three of the twelve residues (GLU7, TYR10, LYS46), the calculations gave positive pKa shift whereas the actual shift is negative. However, the calculations do fairly well in capturing the three largest pKa shifts including the negative shift $\Delta pK_a^{0,exp} = -0.8$ for ASP50, and two positive shifts $\Delta pK_a^{0,exp} = +1.4$ for TYR23, $\Delta pK_a^{0,exp} = +1.0$ for TYR35.

For OMTKY3, the model pKa is closest to experiment for five sites, while the intrinsic pKa and predicted pKa are closest for four sites and three sites, respectively. The RMSDs are higher than for BPTI, but the trends are similar. The model pKa has the smallest RMSD at slightly more than 1 pK unit. At the lower dielectric constant $\epsilon_p = 4$, the RMSD for the intrinsic pKa and predicted pKa is 2.5 pK units or more, while at the higher value $\epsilon_p = 20$ they improve to less than 1.4 pK units, but remain higher than the model pKa RMSD. In this case, the predicted pKa has slightly smaller RMSD than the intrinsic pKa. For five of the twelve residues (ASP27, GLU10, GLU19, LYS13, LYS34), the calculations gave positive pKa shifts whereas the actual shifts are negative. Among the five residues with pKa shifts greater than 1 pK unit, the calculations fail to capture two negative shifts $\Delta pK_a^{0,exp} = -1.8$ for ASP27, $\Delta pK_a^{0,exp} = -1.2$ for GLU19; however, the calculations do fairly well in capturing the negative shift $\Delta pK_a^{0,exp} = -1.6$ for

Table 3. Numerical results for BPTI, OMTKY3, protein dielectric constant $\epsilon_p = 4, 20$, residue ID, model pKa (pK_a^0), experimental pKa (pK_a^{exp}), pKa shift ($\Delta pK_a^{0,exp}$), intrinsic pKa (pK_a^{int}), predicted pKa (pK_a^{pre}), bold value is closest to pK_a^{exp} , RMSD is with respect to pK_a^{exp} .¹²

BPTI ID	pK_a^0	pK_a^{exp}	$\Delta pK_a^{0,exp}$	$\epsilon_p = 4$	$\epsilon_p = 20$	$\epsilon_p = 4$	$\epsilon_p = 20$
				pK_a^{int}		pK_a^{pre}	
ASP3	4.0	3.6	-0.4	3.61	3.89	3.88	3.96
ASP50	4.0	3.2	-0.8	0.22	3.23	3.08	3.87
GLU7	4.4	3.9	-0.5	5.38	4.66	5.84	4.69
GLU49	4.4	4.0	-0.4	3.87	4.37	5.05	4.70
TYR10	9.6	9.4	-0.2	12.01	10.38	12.23	10.46
TYR21	9.6	10.0	+0.4	11.23	10.15	13.04	10.51
TYR23	9.6	11.0	+1.4	16.64	11.24	17.46	11.42
TYR35	9.6	10.6	+1.0	10.11	9.84	10.19	9.86
LYS15	10.4	10.4	+0.0	10.38	10.61	10.32	10.47
LYS26	10.4	10.1	-0.3	10.29	10.54	10.16	10.39
LYS41	10.4	10.6	+0.2	11.65	11.31	12.88	11.44
LYS46	10.4	9.9	-0.5	11.61	11.16	11.15	10.60
RMSD	0.63			2.15	0.63	2.43	0.65
OMTKY3							
ASP7	4.0	2.4	-1.6	1.31	3.55	2.26	3.82
ASP27	4.0	2.2	-1.8	10.20	5.70	6.26	4.90
GLU10	4.4	4.1	-0.3	4.89	4.91	7.09	5.36
GLU19	4.4	3.2	-1.2	5.76	4.80	8.25	5.26
GLU43	4.4	4.8	+0.4	4.99	4.72	4.95	4.69
HIS52	6.3	7.5	+1.2	7.54	7.27	7.94	7.30
TYR11	9.6	10.2	+0.6	13.30	10.68	13.31	10.76
TYR20	9.6	11.1	+1.5	11.49	10.29	12.79	10.78
LYS13	10.4	9.9	-0.5	12.16	11.63	12.20	11.48
LYS29	10.4	11.1	+0.7	11.38	11.24	11.80	11.30
LYS34	10.4	10.1	-0.3	12.53	11.66	12.32	11.45
LYS55	10.4	11.1	+0.7	11.77	11.36	12.10	11.28
RMSD	1.04			2.79	1.39	2.50	1.29

ASP7, and two positive shifts $\Delta pK_a^{0,\text{exp}} = +1.2$ for HIS52, $\Delta pK_a^{0,\text{exp}} = +1.5$ for TYR20.

It is noteworthy that in terms of RMSD, the model pKa does as well or better than the predicted pKa, emphasizing the limitations of continuum dielectric models for predicting protein pKas.^{53,54} This could possibly be remedied by including nonpolar effects or spatial variation of the protein dielectric constant. Other improvements suggested in the literature call for including the effect of protein conformational flexibility,^{55,56} replacing the discontinuous dielectric interface with a smooth Gaussian function to partially represent solvent penetration^{57,58} and using microscopic models to account for atomic polarization and dipole reorientation.^{59,60} In Table 3 the higher protein dielectric constant $\epsilon_p = 20$ yields smaller RMSD than the lower value $\epsilon_p = 4$, as found in several previous studies.^{13,61,62} Nonetheless, for some residues, $\epsilon_p = 4$ yields better agreement with experiment than $\epsilon_p = 20$ (ASP3 in BPTI, HIS52 in OMTKY3); this is consistent with previous results showing that the best choice of protein dielectric constant for a given residue can depend on the residue's exposure to solvent.⁶³ Overall, the present results indicate the complexity of these pH-dependent processes and they support the need for further investigations to develop a more physically realistic yet computationally tractable model.

To further test our pKa computing procedure, we chose five other proteins from the pKa database collected by Pahari *et al.*⁶⁴ The database contains more than 200 proteins with experimental pKas and our selection is due to the efficiency limitations of our current procedure which solves $O(N_t^2)$ PB equations and involves $O(2^{N_t})$ titrating site-site interactions; a significantly faster version of TABI and a more efficient

strategy to accelerate the site-site interactions are under development. Table 4 reports the RMSD of the intrinsic pKa, predicted pKa, and model pKa compared with experimental pKas⁶⁴ for each protein and for the total set of residues. From this table, we observe similar trends as seen above with BPTI and OMTKY3, namely that including site-site interactions in the predicted pKa do not significantly improve upon the intrinsic pKa, and the computed pKa (intrinsic or predicted) on average is not as good as the model pKa in approaching the experimental pKa. This reinforces the need to improve the current continuum dielectric model along the lines mentioned above.

7. CONCLUSION

A common approach to computing protein pKas uses a continuum dielectric model in which the protein is a low dielectric medium with embedded atomic point charges, the solvent is a high dielectric medium with a Boltzmann distribution of ionic charges, and the pKa is related to the electrostatic free energy which is obtained by solving the PB equation. Starting from the model pKa for a titrating residue, the method obtains the intrinsic pKa and then computes the protonation probability for a given pH including site-site interactions.^{5,13} This approach assumes that acid dissociation does not affect protein conformation aside from adding or deleting charges at titratable sites. In this work, we demonstrated our TABI solver for the relevant electrostatic calculations.³⁵ The TABI code accurately resolves the complex charge distributions of protonated and deprotonated states using the appropriate analytic Green's functions. The pKa computing procedure is enclosed in a convenient Python wrapper which is publicly available at the corresponding author's website.

Results were presented for several proteins compared with experimental data. In terms of RMSD with respect to experimental values for the entire set of titratable sites on each protein, the model pKa does as well or better than the intrinsic pKa and predicted pKa. Nonetheless, the calculations did succeed for several residues with the largest pKa shifts from the model pKa. Among ongoing efforts to improve protein pKa calculations,⁶ the advantage of TABI is that it reduces the numerical errors in the electrostatic calculations so that attention can be focused on modeling assumptions. Further improvements to TABI are expected in future work using our recently developed GPU-accelerated barycentric treecode.^{65,66}

Table 4. Results for five proteins selected from a pKa database,⁶⁴ protein dielectric constant $\epsilon_p = 20$, column 1 (PDB ID), column 2 (number of atoms), column 3 (number of residues for which experimental pKas are known), columns 4, 5, 6 (RMSD for intrinsic pKa^{int}, predicted pKa^{pre}, model pKa⁰, compared with experimental pKa^{exp}), last row gives results for all five proteins.

PDB ID	N_{atom}	N_{res}	pKa ^{int,exp}	pKa ^{pre,exp}	pKa ^{0,exp}
1PGA	436	13	0.85	0.65	0.51
2QMT	438	13	1.30	1.39	0.51
2CI2	521	10	1.18	1.52	0.84
1IGD	468	10	1.08	1.19	0.46
1BPI	460	8	0.94	1.03	0.56
Total	—	54	1.09	1.19	0.59

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation grants DMS-1819094 and DMS-1819193. J. Hu and Y. Xu were partially supported by the Hamilton Scholarship from Southern Methodist University.

References

- Ullmann, G. M.; Knapp, E. W. Electrostatic Models for Computing Protonation and Redox Equilibria in Proteins. *Eur. Biophys. J.* **1999**, *28*, 533–551.
- Onufriev, A. V.; Alexov, E. Protonation and pK Changes in Protein-Ligand Binding. *Q. Rev. Biophys.* **2013**, *46*, 181–209.
- Nielsen, J. E. Analyzing Protein NMR pH-Titration Curves, in *Annual Reports in Computational Chemistry*, eds., Wheeler, R. A.; Spellmeyer, D. C. Elsevier **2008**, pp. 89–106.
- Lee, A. C.; Crippen, G. M. Predicting pKa. *J. Chem. Inf. Model.* **2009**, *49*, 2013–2033.
- Bashford, D.; Karplus, M. pK_a's of Ionizable Groups in Proteins: Atomic Detail from a Continuum Electrostatic Model. *Biochemistry* **1990**, *29*, 10219–10225.
- Alexov, E.; Mehler, E. L.; Baker, N.; Baptista, A. M.; Huang, Y.; Milletti, F.; Nielsen, J. E.; Farrell, D.; Carstensen, T.; Olsson, M. H. M.; Shen, J. K.; Warwicker, J.; Williams, S.; Word, J. M. Progress in the Prediction of pKa Values in Proteins. *Proteins* **2011**, *79*, 3260–3275.
- Antosiewicz, J. M.; Shugar, D. Poisson-Boltzmann Continuum-Solvation Models: Applications to pH-Dependent Properties of Biomolecules. *Mol. BioSyst.* **2011**, *7*, 2923–2949.
- Kilambi, K. P.; Gray, J. J. Rapid Calculation of Protein pKa Values Using Rosetta. *Biophys. J.* **2012**, *103*, 587–595.
- Meyer, T.; Knapp, E. W. pKa Values in Proteins Determined by Electrostatics Applied to Molecular Dynamics Trajectories. *J. Chem. Theory Comput.* **2015**, *11*, 2827–2840.
- Riccardi, D.; Schaefer, P.; Cui, Q. pKa Calculations in Solution and Proteins with QM/MM Free Energy Perturbation Simulations: A Quantitative Test of QM/MM Protocols. *J. Phys. Chem. B* **2005**, *109*, 17715–17733.
- Gosink, L. J.; Hogan, E. A.; Pulsipher, T. C.; Baker, N. A. Bayesian Model Aggregation for Ensemble-Based Estimates of Protein pKa Values. *Proteins* **2014**, *82*, 354–363.
- Li, H.; Robertson, A. D.; Jensen, J. H. Very Fast Empirical Prediction and Rationalization of Protein pKa Values. *Proteins* **2005**, *61*, 704–721.
- Juffer, A. H.; Argos, P.; Vogel, H. J. Calculating Acid-Dissociation Constants of Proteins using the Boundary Element Method. *J. Phys. Chem. B* **1997**, *101*, 7664–7673.
- Ho, K. L. *Fast Direct Methods for Molecular Electrostatics*. PhD Thesis, New York University, **2012**.
- Fogolari, F.; Brigo, A.; Molinari, H. The Poisson-Boltzmann Equation for Biomolecular Electrostatics: A Tool for Structural Biology. *J. Mol. Recognit.* **2002**, *15* (6), 377–92.
- Baker, N. A. Poisson-Boltzmann Methods for Biomolecular Electrostatics. *Methods Enzymol.* **2004**, *383*, 94–118.
- Connolly, M. L. Depth Buffer Algorithms for Molecular Modelling. *J. Mol. Graphics* **1985**, *3*, 19–24.
- Sanner, M. F.; Olson, A. J.; Spehner, J. C. Reduced Surface: An Efficient Way to Compute Molecular Surfaces. *Biopolymers* **1996**, *38*, 305–320.
- Geng, W. A Boundary Integral Poisson-Boltzmann Solvers Package for Solvated Bimolecular Simulations. *Comput. Math. Biophys.* **2015**, *3*, 43–58.
- Holst, M. J. *The Poisson-Boltzmann Equation: Analysis and Multilevel Numerical Solution*. PhD Thesis, UIUC, **1994**.
- Lu, B. Z.; Zhou, Y. C.; Holst, M. J.; McCammon, J. A. Recent Progress in Numerical Methods for the Poisson-Boltzmann Equation in Biophysical Applications. *Comm. Comput. Phys.* **2008**, *3*(5), 973–1009.
- Baker, N. A. Improving Implicit Solvent Simulations: A Poisson-Centric View. *Curr. Opin. Struct. Biol.* **2005**, *15* (2), 137–143.
- Yang, A. S.; Gunner, M. R.; Sampogna, R.; Sharp, K.; Honig, B. On the Calculation of pKas in Proteins. *Proteins Struct. Funct. Genet.* **1993**, *15*, 252–265.
- Holst, M. J.; Saied, F. Numerical Solution of the Non-linear Poisson-Boltzmann Equation: Developing more Robust and Efficient Methods. *J. Comput. Chem.* **1995**, *16*, 337–364.
- Honig, B.; Nicholls, A. Classical Electrostatics in Biology and Chemistry. *Science* **1995**, *268*, 1144–1149.
- Im, W.; Beglov, D.; Roux, B. Continuum Solvation Model: Electrostatic Forces from Numerical Solutions to the Poisson-Boltzmann Equation. *Comput. Phys. Commun.* **1998**, *111*, 59–75.
- Baker, N. A.; Sept, D.; Joseph, S.; Holst, M. J.; McCammon, J. A. Electrostatics of Nanosystems: Application to Microtubules and the Ribosome. *Proc. Natl. Acad. Sci.* **2001**, *98*, 10037–10041.
- Luo, R.; David, L.; Gilson, M. K. Accelerated Poisson-Boltzmann Calculations for Static and Dynamic Systems. *J. Comput. Chem.* **2002**, *23*, 1244–1253.
- Chen, D.; Chen, Z.; Chen, C.; Geng, W.; Wei, G. W. MIBPB: A Software Package for Electrostatic Analysis. *J. Comput. Chem.* **2011**, *32*, 657–670.
- Deng, W.; Zhufu, X.; Xu, J.; Zhao, S. A New Discontinuous Galerkin Method for the Nonlinear Poisson-Boltzmann Equation. *Appl. Math. Lett.* **2015**, *257*, 1000–1021.
- Zauhar, R. J.; Morgan, R. S. A New Method for Computing the Macromolecular Electric Potential. *J. Mol. Biol.* **1985**, *186*, 815–820.

32. Yoon, B. J.; Lenhoff, A. M. A Boundary Element Method for Molecular Electrostatics with Electrolyte Effects. *J. Comput. Chem.* **1990**, *11*, 1080–1086.
33. Juffer, A.; Botta, E. F. F.; van Keulen, B.; van der Ploeg, A.; Berendsen, H. The Electric Potential of a Macromolecule in a Solvent: A Fundamental Approach. *J. Comput. Phys.* **1991**, *97*, 144–171.
34. Boschitsch, A. H.; Fenley, M. O.; Zhou, H. X. Fast Boundary Element Method for the Linear Poisson-Boltzmann Equation. *J. Phys. Chem. B* **2002**, *106*(10), 2741–2754.
35. Geng, W.; Krasny, R. A Treecode-Accelerated Boundary Integral Poisson-Boltzmann Solver for Electrostatics of Solvated Biomolecules. *J. Comput. Phys.* **2013**, *247*, 62–78.
36. Zhang, B.; Lu, B.; Cheng, X.; Huang, J.; Pitsianis, N. P.; Sun, X.; McCammon, J. A. Mathematical and Numerical Aspects of the Adaptive Fast Multipole Poisson-Boltzmann Solver. *Commun. Comput. Phys.* **2013**, *13*, 107–128.
37. Zhong, Y.; Ren, K.; Tsai, R. An Implicit Boundary Integral Method for Computing Electric Potential of Macromolecules in Solvent. *J. Comput. Phys.* **2018**, *359*, 199–215.
38. Cancès, E.; Maday, Y.; Stamm, B. Domain Decomposition for Implicit Solvation Models. *J. Chem. Phys.* **2013**, *139*(5), 054111.
39. Saad, Y.; Schultz, M. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. Stat. Comput.* **1986**, *7*, 856–869.
40. Barnes, J.; Hut, P. A Hierarchical O(NlogN) Force-Calculation Algorithm. *Nature* **1986**, *324*, 446–449.
41. Li, P.; Johnston, H.; Krasny, R. A Cartesian Treecode for Screened Coulomb Interactions. *J. Comput. Phys.* **2009**, *228*, 3858–3868.
42. Marquart, M.; Walter, J.; Deisenhofer, J.; Bode, W.; Huber, R. The Geometry of the Reactive Site and of the Peptide Groups in Trypsin, Trypsinogen and its Complexes with Inhibitors. *Acta Cryst. Sec. B* **1983**, *39*(4), 480–490.
43. Bode, W.; Epp, O.; Huber, R.; Laskowski, M.; Ardelt, W. The Crystal and Molecular Structure of the Third Domain of Silver Pheasant Ovomuroid (omsvp3). *Eur. J. Biochem.* **1985**, *147*(2), 387–395.
44. Dolinsky, T. J.; Czodrowski, P.; Li, H.; Nielsen, J. E.; Jensen, J. H.; Klebe, G.; Baker, N. A. PDB2PQR: Expanding and Upgrading Automated Preparation of Biomolecular Structures for Molecular Simulations. *Nucleic Acids Res.* **2007**, *35*, W522–W525.
45. Sitkoff, D.; Sharp, K. A.; Honig, B. Accurate Calculation of Hydration Free Energies Using Macroscopic Solvent Models. *J. Phys. Chem.* **1994**, *98*, 1978–1988.
46. Chen, J.; Geng, W. On Preconditioning the Treecode-Accelerated Boundary Integral (TABI) Poisson-Boltzmann Solver. *J. Comput. Phys.* **2018**, *373*, 750–762.
47. Warshel, A. Calculations of Enzymatic Reactions: Calculations of pKa, Proton Transfer Reactions, and General Acid Catalysis Reactions in Enzymes. *Biochemistry* **1981**, *20*, 3167–3177.
48. Nozaki, Y.; Tanford, C. Examination of Titration Behaviour. *Methods Enzymol.* **1967**, *11*, 715–734.
49. Hu, J.; Zhao, S.; Geng, W. Accurate pKa Computation using Matched Interface and Boundary (MIB) Method Based Poisson-Boltzmann Solver. *Commun. Comput. Phys.* **2018**, *23*, 520–539.
50. Bashford, D.; Karplus, M. Multiple-Site Titration Curves of Proteins: An Analysis of Exact and Approximate Methods for their Calculation. *J. Phys. Chem.* **1991**, *95*, 9556–9561.
51. Beroza, P.; Fredkin, D. R.; Okamura, M. Y.; Feher, G. Protonation of Interacting Residues in a Protein by a Monte Carlo Method: Application to Lysozyme and the Photosynthetic Reaction Center of *Rhodobacter sphaeroides*. *Proc. Natl. Acad. Sci.* **1991**, *88*, 5804–5808.
52. Myers, J.; Grothaus, G.; Narayanan, S.; Onufriev, A. A Simple Clustering Algorithm can be Accurate Enough for Use in Calculations of pKs in Macromolecules. *Proteins* **2006**, *63*, 928–938.
53. Anandakrishnan, R.; Aguilar, B.; Onufriev, A. V. H++3.0: Automating pK Prediction and the Preparation of Biomolecular Structures for Atomistic Molecular Modeling and Simulations. *Nucleic Acids Res.* **2012**, *40*, W537–W541.
54. Gunner, M. R.; Baker, N. A. Continuum Electrostatics Approaches to Calculating pKas and Ems in Proteins, in *Computational Approaches for Studying Enzyme Mechanism Part B*, ed., Voth, G. A. Academic Press **2016**, pp. 1–20.
55. Alexov, E. G.; Gunner, M. R. Incorporating Protein Conformational Flexibility into the Calculation of pH-Dependent Protein Properties. *Biophys. J.* **1997**, *74*, 2075–2093.
56. Georgescu, R. E.; Alexov, E. G.; Gunner, M. R. Combining Conformational Flexibility and Continuum Electrostatics for Calculating pKas in Proteins. *Biophys. J.* **2002**, *83*, 1731–1748.
57. Word, J. M.; Nicholls, A. Application of the Gaussian Dielectric Boundary in Zap to the Prediction of Protein pKa Values. *Proteins* **2011**, *79*, 3400–3409.
58. Wang, L.; Li, L.; Alexov, E. pKa Predictions for Proteins, RNAs, and DNAs with the Gaussian Dielectric Function using DelPhiPKa. *Proteins* **2015**, *83*, 2186–2197.
59. Warshel, A.; Levitt, M. Theoretical Studies of Enzymic Reactions: Electrostatic and Steric Stabilization of the Carbonium Ion in the Reaction of Lysozyme. *J. Mol. Biol.* **1976**, *103*, 227–249.
60. Sham, Y. Y.; Chu, Z. T.; Warshel, A. Consistent Calculations of pKa's of Ionizable Residues in Proteins: Semi-Microscopic and Microscopic Approaches. *J. Phys. Chem. B* **1997**, *101*, 4458–4472.

61. Antosiewicz, J.; McCammon, J. A.; Gilson, M. K. Prediction of pH-Dependent Properties of Proteins. *J. Mol. Biol.* **1994**, 238, 415–436.
62. Antosiewicz, J.; McCammon, J. A.; Gilson, M. K. The Determinants of pKas in Proteins. *Biochemistry* **1996**, 35, 7819–7833.
63. Demchuk, E.; Wade, R. Improving the Continuum Dielectric Approach to Calculating pKas of Ionizable Groups in Proteins. *J. Phys. Chem.* **1996**, 100, 17373–17387.
64. Pahari, S.; Sun, L.; Alexov, E. PKAD: A Database of Experimentally Measured pKa Values of Ionizable Groups in Proteins. *Database J. Biol. Database Curation* **2019**, 2019, article ID baz024.
65. Wang, L.; Krasny, R.; Tlupova, S. A Kernel-Independent Treecode Based on Barycentric Lagrange Interpolation. *Commun. Comput. Phys.* **2020**, 28, 1415–1436.
66. Vaughn, N.; Wilson, L.; Krasny, R. A GPU-Accelerated Barycentric Lagrange Treecode, in *2020 IEEE Int. Parallel and Distributed Processing Symp. Workshop (IPDPSW)* IEEE, **2020**, pp. 701–710.