Corrected Trapezoidal Rule-IBIM for linearized Poisson-Boltzmann equation

Federico Izzo,† Yimin Zhong‡ Olof Runborg§ Richard Tsai¶

Abstract

In this paper, we solve the linearized Poisson-Boltzmann equation, used to model the electric potential of macromolecules in a solvent. We derive a corrected trapezoidal rule with improved accuracy for a boundary integral formulation of the linearized Poisson-Boltzmann equation. More specifically, in contrast to the typical boundary integral formulations, the corrected trapezoidal rule is applied to integrate a system of compacted supported singular integrals using uniform Cartesian grids in \( \mathbb{R}^3 \), without explicit surface parameterization. A Krylov method, accelerated by a fast multipole method, is used to invert the resulting linear system. We study the efficacy of the proposed method, and compare it to an existing, lower order method. We then apply the method to the computation of electrostatic potential of macromolecules immersed in solvent. The solvent excluded surfaces, defined by a common approach, are merely piecewise smooth, and we study the effectiveness of the method for such surfaces.

Key words: Poisson-Boltzmann equation; implicit boundary integral method; implicit solvent model; singular integrals; trapezoidal rules.

AMS subject classifications 2020: 45A05, 65R20, 6N5D30, 65N80, 78M16, 92E10

1 Introduction

We propose to solve the linearized Poisson-Boltzmann equation using an Implicit Boundary Integral Method (IBIM), discretized by a corrected trapezoidal rule (CTR). The linearized Poisson-Boltzmann equation can be used to model the electrostatics of charged macromolecule-solvent systems. Numerical modeling and simulation of such systems is an active and relevant research topic. Electrostatic interactions with the solvent significantly affect the overall macromolecule behavior. They are relevant for example in electrochemistry, as an aqueous solvent plays a significant role in the dynamical processes of biological molecules (see [1, 4] for comprehensive introductions to the problem).

When describing the electrostatic interactions, implicit solvent methods approximate the problem by simplifying the description of the solvent environment and consequently greatly reducing the degrees of freedom. The frameworks for approximating implicitly the electrostatic interactions are for example the Coulomb-field approximation, the generalized Born models, and the Poisson-Boltzmann theory. The latter two are more popular, and among them the Poisson-Boltzmann theory offers a more detailed representation at the cost of more expensive computations (see also [3]).

The Poisson-Boltzmann equation, which describes the electrostatic field determined by the interactions in the Poisson-Boltzmann theory, is nonlinear. The solution to its linearized form is however widely used, both because it represents a valid approximation to the nonlinear solution in certain settings, and because it can be used iteratively to find the original nonlinear solution.
Hence, numerically solving the linearized Poisson-Boltzmann equation is still an active and relevant problem (see e.g. [11, 7, 18, 23, 22]).

In this work, we propose a fast numerical method for solving the linearized Poisson-Boltzmann equation on surfaces:

\[ -\nabla \cdot (\varepsilon_I \nabla \psi(x)) = \sum_{k=1}^{N_x} q_k \delta(x - z_k), \quad \text{in } \Omega, \]
\[ -\nabla \cdot (\varepsilon_E \nabla \psi(x)) = -\kappa \psi(x), \quad \text{in } \Omega^c, \]
\[ \psi(x)|_{\Gamma^-} = \psi(x)|_{\Gamma^+}, \quad \text{on } \Gamma, \]
\[ \varepsilon_I \frac{\partial \psi}{\partial n}|_{\Gamma^-} = \varepsilon_E \frac{\partial \psi}{\partial n}|_{\Gamma^+}, \quad \text{on } \Gamma, \]
\[ |x| \psi(x) \to 0, \quad |x|^2 |\nabla \psi(x)| \to 0, \quad \text{as } |x| \to \infty. \tag{1.1} \]

In (1.1), \( \psi \) represents the electrostatic potential, \( N_x \) is the number of atoms composing the macromolecules, which have centers \( \{z_k\}_{k=1}^{N_x} \), radii \( \{r_k\}_{k=1}^{N_x} \), and charge numbers \( \{q_k\}_{k=1}^{N_x} \) respectively. \( \Gamma \) represents the closed surface which separates the region occupied by the macromolecule (here denoted by \( \Omega \)) and the rest of the space \( \Omega^c := \mathbb{R}^3 \setminus \Omega = \mathbb{R}^3 \setminus (\Omega \cup \Gamma) \). We will describe in Section 2.6 how we test the method on simple smooth surfaces \( \Gamma \), and then in Section 3 how we apply it to actual solvent-molecule interfaces. The operator \( \frac{\partial}{\partial n}(x) \) represents the normal derivative of \( \psi \) in \( x \in \Gamma \) with normal pointing outward from \( \Gamma \). The parameters \( \varepsilon_I \) and \( \varepsilon_E \) are the dielectric constants inside and outside \( \Omega \) respectively, and \( \kappa \) is a screening parameter. The radiation conditions on the last row are needed to ensure uniqueness of the solutions.

We use the following boundary integral formulation [11] to find the solution to the equations (1.1):

\[ \frac{1}{2} \left( 1 + \frac{\varepsilon_E}{\varepsilon_I} \right) \psi(x) + \int_{\Gamma} K_{11}(x, y) \psi(y) dy = \int_{\Gamma} K_{12}(x, y) \frac{\partial \psi}{\partial n}(y) dy = \sum_{k=1}^{N_x} \frac{q_k}{\varepsilon_I} G_0(x, z_k), \tag{1.2} \]
\[ \frac{1}{2} \left( 1 + \frac{\varepsilon_I}{\varepsilon_E} \right) \frac{\partial \psi}{\partial n}(x) + \int_{\Gamma} K_{21}(x, y) \psi(y) dy = \int_{\Gamma} K_{22}(x, y) \frac{\partial \psi}{\partial n}(y) dy = \sum_{k=1}^{N_x} \frac{q_k}{\varepsilon_I} \frac{\partial G_0}{\partial n_x}(x, z_k). \tag{1.3} \]

where

\[
K_{11}(x, y) := \frac{\partial G_0}{\partial n_y}(x, y) - \frac{\varepsilon_E}{\varepsilon_I} \frac{\partial G_{\kappa}}{\partial n_y}(x, y), \quad K_{12}(x, y) := G_0(x, y) - G_\kappa(x, y),
\]
\[
K_{21}(x, y) := \frac{\partial^2 G_0}{\partial n_x \partial n_y}(x, y) - \frac{\partial^2 G_{\kappa}}{\partial n_x \partial n_y}(x, y), \quad K_{22}(x, y) := \frac{\partial G_0}{\partial n_x}(x, y) - \frac{\varepsilon_I}{\varepsilon_E} \frac{\partial G_{\kappa}}{\partial n_x}(x, y).
\tag{1.4}
\]

The fundamental solutions which define the kernels in (1.4) are

\[
G_0(x, y) := \frac{1}{4\pi|x - y|}, \quad G_\kappa(x, y) := \frac{e^{-\kappa |x - y|}}{4\pi|x - y|}. \tag{1.5}
\]

The integrals in (1.2)-(1.3) are well defined because of the integrability of the kernels for \( x = y \):

\[
K_{11}(x, y) \sim O(|x - y|^{-1}), \quad K_{21}(x, y) \sim O(|x - y|^{-1}), \quad K_{22}(x, y) \sim O(|x - y|^{-1}), \quad K_{12}(x, y) \sim O(1).
\tag{1.6}
\]

As in [22], we write these integrals in the Implicit Boundary Integral Method (IBIM) framework (see [13, 12]).

We point out that the kernel \( K_{21} \) is integrable even though it is defined as the difference between two hypersingular kernels which are not Cauchy integrable. This is the power of the boundary integral formulation derived by Juffer et al. [11]. The kernel \( K_{21} \) proves to be the bottleneck of the accuracy order attainable given a second order approximation of the surface around the target point \( x \). If the approximation is improved to third order, it is possible to improve the order of accuracy for the kernels \( K_{11}, K_{22}, \) and \( K_{12} \) but not for \( K_{21} \). Only a fourth order approximation allows us to increase the order of accuracy for \( K_{21} \). This will be explained in more detail in Section 2.4.1.
2 Numerical solution of the volumetric boundary integral formulation

In this section we will provide the necessary background information for the numerical solution of (1.2)-(1.3).

In Section 2.1 we present the volumetric formulation of the surface integrals via non-parametrized surface representation. In Section 2.2 we present the discretization (2.14)-(2.15) of the original BIEs (1.2)-(1.3) using the framework shown in the previous subsection. In Section 2.3 we briefly present the kernel regularization (K-reg) developed in [22]. In Section 2.4 we present the formally second order accurate quadrature rule (CTR2) based on the trapezoidal rule which we use to approximate the singular volume integrals. In Section 2.5 we present an overview of the algorithms used to apply the quadrature rule given the surface $\Gamma$. In Section 2.6 we present numerical tests on smooth surfaces to show the order of accuracy of CTR2 and compare it to K-reg.

2.1 Extensions of the singular boundary integral operators

Let $\Omega \subset \mathbb{R}^3$ be a bounded open set with $C^2$ boundaries, and $\partial \Omega =: \Gamma$. We shall refer to $\Gamma$ as the surface. Let $f$ be a function defined on $\Gamma$. In this section we present an approach for extending a boundary integral

$$\int_{\Gamma} f(y) d\sigma_y,$$  \hspace{1cm} (2.1)

to a volumetric integral around the surface. Instead of parameterizations, this approach relies on the Euclidean distance to the surface, and its derivatives. More precisely, we define the signed distance function

$$d_{\Gamma}(x) := \begin{cases} \min_{y \in \Gamma} \| x - y \|, & \text{if } x \in \Omega \\ -\min_{y \in \Gamma} \| x - y \|, & \text{if } x \in \Omega^c \end{cases}$$  \hspace{1cm} (2.2)

and the closest point projection

$$P_{\Gamma}(x) := \arg\min_{y \in \Gamma} \| x - y \|. \hspace{1cm} (2.3)$$

If there is more than one global minimum, we pick one randomly from the set. Let $C_{\Gamma}$ denote the set of points in $\mathbb{R}^3$ which are equidistant to at least two distinct points on $\Gamma$. The reach $\tau$ is defined as $\inf_{x \in \Gamma, y \in C_{\Gamma}} \| x - y \|$. Clearly, $\tau$ is restricted by the local geometry (the curvatures) and the global structure of $\Gamma$ (the Euclidean and geodesic distances between any two points on $\Gamma$).

In this paper, we assume that $\Gamma$ is $C^2$ and thus has a non-zero reach. Let $T_{\varepsilon}$ denote the set of points of distance at most $\varepsilon$ from $\Gamma$:

$$T_{\varepsilon} = \{ x \in \mathbb{R}^3 : |d_{\Gamma}(x)| \leq \varepsilon \}. \hspace{1cm} (2.4)$$

Then $P_{\Gamma}$ is a diffeomorphism between $\Gamma$ and the level sets of $d_{\Gamma}$ in $T_{\varepsilon}$. Furthermore,

$$P_{\Gamma}(x) = x - d_{\Gamma}(x) \nabla d_{\Gamma}(x), \hspace{0.5cm} x \in T_{\varepsilon}. \hspace{1cm}$$

We define the extension of the integrand $f$ by

$$\overline{f}(x) := f(P_{\Gamma}(x)), \hspace{0.5cm} x \in \mathbb{R}^3. \hspace{1cm} (2.5)$$

As in [12, 13], we can then rewrite the surface integral (2.1):

$$\int_{\Gamma} f(y) d\sigma_y = \int_{\mathbb{R}^3} \overline{f}(x) \delta_{\Gamma, \varepsilon}(y) dy = \int_{T_{\varepsilon}} \overline{f}(x) \delta_{\Gamma, \varepsilon}(y) dy,$$  \hspace{1cm} (2.6)

where

$$\delta_{\Gamma, \varepsilon}(y) := J_{\Gamma}(y) \delta_{\varepsilon}(d_{\Gamma}(y)), \hspace{0.5cm} y \in \mathbb{R}^3,$$

$$\delta_{\varepsilon}(\eta) := \frac{1}{\varepsilon} \phi \left( \frac{\eta}{\varepsilon} \right), \hspace{0.5cm} \phi \in C^\infty(\mathbb{R}) \text{ supported in } [-1, 1], \hspace{0.5cm} \text{and } \int_{\mathbb{R}} \phi(x) dx = 1, \hspace{1cm} (2.7)$$
and $J_T(y')$ is the Jacobian of the transformation from $\Gamma$ to the level set $\Gamma_\eta := \{y \in \mathbb{R}^n : d_T(y) = \eta\}$. In $\mathbb{R}^4$, $J_T(y')$ is a quadratic polynomial in $d_T(y')$:

$$J_T(y') := 1 + 2d_T(y')\mathcal{H}(y') + d_T(y')^2\mathcal{G}(y').$$  \hspace{1cm} (2.8)

where $\mathcal{H}(y')$ and $\mathcal{G}(y')$ are respectively the mean and Gaussian curvatures of $\Gamma_\eta$ at $y'$. The curvatures as well as the corresponding principal directions can be extracted from $P_1$; see [13] for more detail.

Our primary focus is when $f(y)$ is replaced by a function of the kind $K(x, y)\zeta(y)$, corresponding to the kernels and potentials found in (1.2)-(1.3), (1.4)), and the integral operators of form

$$\mathcal{J}_i[\zeta_1, \zeta_2](x) := \int_\Gamma K_{1i}(x, y)\zeta_1(y)dy - \int_\Gamma K_{2i}(x, y)\zeta_2(y)dy, \quad x \in \mathbb{R}^3, \quad i = 1, 2. \hspace{1cm} (2.9)$$

We can now write the operators (2.9) in volumetric form:

$$\mathcal{J}_i[\rho_1, \rho_2](x) := \int_{T_\epsilon} \mathcal{K}_{1i}(x, y)\rho_1(y)\delta_{\epsilon, x}(y)dy - \int_{T_\epsilon} \mathcal{K}_{2i}(x, y)\rho_2(y)\delta_{\epsilon, x}(y)dy, \quad x \in T_\epsilon, \quad i = 1, 2, \hspace{1cm} (2.10)$$

where

$$\mathcal{K}(x, y) := K(x, P_1 y), \quad x, y \in \mathbb{R}^n. \hspace{1cm} (2.11)$$

It is important to notice that if $K(x, y)$ is singular for $x = y$, then $\mathcal{K}(P_1 x, y)$ is singular on the set

$$\{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n : P_1 x = P_1 y\},$$

i.e. for a fixed $x^* \in \Gamma$, $\mathcal{K}(x^*, y)$ is singular along the normal line passing through $x^*$, while $K(x^*, y)$ is singular in a point.

Finally, with

$$\lambda_1 := \frac{1}{2} \left(1 + \frac{\varepsilon_I}{\varepsilon_E}\right), \quad \lambda_2 := \frac{1}{2} \left(1 + \frac{\varepsilon_I}{\varepsilon_E}\right), \quad g_1(x) := \sum_{k=1}^{N_c} \frac{q_k}{\varepsilon_I} G_0(x, z_k), \quad g_2(x) := \sum_{k=1}^{N_c} \frac{q_k}{\varepsilon_I} \frac{\partial G_0}{\partial n_x}(x, z_k), \hspace{1cm} (2.12)$$

the volumetric forms of equations (1.2)-(1.3) are derived:

$$\lambda_i \rho_i(x) + \mathcal{J}_i[\rho_1, \rho_2](P_1 x) = g_i(P_1 x), \quad x \in T_\epsilon, \quad i = 1, 2. \hspace{1cm} (2.13)$$

The solutions $\rho_1$, $\rho_2$ will coincide with the constant extensions along the normals of $\psi$ and $\psi_n$ respectively: $\rho_1(x) \equiv \psi(P_1(x)), \rho_2(x) \equiv \psi_n(P_1(x))$; see, for example, the arguments given in [9].

2.2 Quadrature rules on uniform Cartesian grids

In this paper, we derive numerical quadratures for the singular integral operators $\mathcal{J}_i[\rho_1, \rho_2](x)$ for $x \in \Gamma$ (equivalently, $\mathcal{J}_i[\rho_1, \rho_2](P_1 x)$ for $x \in T_\epsilon$) for $i = 1, 2$. The quadrature rules will be constructed based on the trapezoidal rule for the grid nodes $T_\epsilon^h := T_\epsilon \cap h\mathbb{Z}^3$, which corresponds to the portion of the uniform Cartesian grid $h\mathbb{Z}^3$ within $T_\epsilon$. Since the integrand in (2.10) is singular for $x \in \Gamma$, the trapezoidal rule should be corrected near $x$ for faster convergence. Correction will be defined by summing the judiciously derived weights over a set of grid nodes denoted by $N_h(x)$. The sum will be denoted by $\mathcal{R}_{\epsilon}(x)$. Ultimately, the quadrature for $\mathcal{J}_i[\rho_1, \rho_2](P_1 x)$ will involve the regular Riemann sum of the integrand in $T_\epsilon^h \setminus N_h(x)$, and the correction $\mathcal{R}_{\epsilon}(x)$ in $N_h(x)$:

$$\mathcal{J}_i[\rho_1, \rho_2](x) \approx h^3 \sum_{y_m \in T_\epsilon^h \setminus N_h(x)} \mathcal{K}_{11}(x, y_m)\rho_1(y_m)\delta_{\epsilon, x}(y_m) + h^2 \sum_{y_m \in N_h(x)} \omega_m^{(1)} \rho_1(y_m)\delta_{\epsilon, x}(y_m)$$

$$- h^3 \sum_{y_m \in T_\epsilon^h \setminus N_h(x)} \mathcal{K}_{12}(x, y_m)\rho_2(y_m)\delta_{\epsilon, x}(y_m) - h^2 \sum_{y_m \in N_h(x)} \omega_m^{(11)} \rho_1(y_m)\delta_{\epsilon, x}(y_m),$$

$$- h^3 \sum_{y_m \in T_\epsilon^h \setminus N_h(x)} \mathcal{K}_{21}(x, y_m)\rho_1(y_m)\delta_{\epsilon, x}(y_m) + h^2 \sum_{y_m \in N_h(x)} \omega_m^{(11)} \rho_1(y_m)\delta_{\epsilon, x}(y_m),$$

$$- h^3 \sum_{y_m \in T_\epsilon^h \setminus N_h(x)} \mathcal{K}_{22}(x, y_m)\rho_2(y_m)\delta_{\epsilon, x}(y_m) - h^2 \sum_{y_m \in N_h(x)} \omega_m^{(12)} \rho_2(y_m)\delta_{\epsilon, x}(y_m),$$

$$\omega_m^{(1)} := \frac{1}{2} \left(1 + \frac{\varepsilon_I}{\varepsilon_E}\right), \quad \omega_m^{(11)} := \frac{1}{2} \left(1 + \frac{\varepsilon_I}{\varepsilon_E}\right), \quad \omega_m^{(12)} := \frac{1}{2} \left(1 + \frac{\varepsilon_I}{\varepsilon_E}\right).$$
and
\[ J_2[\rho_1, \rho_2](\mathbf{x}) \approx h^3 \sum_{\mathbf{y}_m \in T^{\rho}_{\delta}(\mathbf{x}) \setminus N_\rho(\mathbf{x})} \mathbf{K}_{21}(\mathbf{x}, \mathbf{y}_m) \rho_1(\mathbf{y}_m) \delta_{\Gamma, \tau}(\mathbf{y}_m) + h^2 \sum_{\mathbf{y}_m \in N_\rho(\mathbf{x})} \omega_{m}^{(21)} \rho_1(\mathbf{y}_m) \delta_{\Gamma, \tau}(\mathbf{y}_m) \]
\[ - h^3 \sum_{\mathbf{y}_m \in T^{\rho}_{\delta}(\mathbf{x}) \setminus N_\rho(\mathbf{x})} \mathbf{K}_{22}(\mathbf{x}, \mathbf{y}_m) \rho_2(\mathbf{y}_m) \delta_{\Gamma, \tau}(\mathbf{y}_m) - h^2 \sum_{\mathbf{y}_m \in N_\rho(\mathbf{x})} \omega_{m}^{(22)} \rho_2(\mathbf{y}_m) \delta_{\Gamma, \tau}(\mathbf{y}_m), \]
where \( \omega_{m}^{(ij)} \) is a weight which depends on the kernel \( K_{ij} \), on the grid node \( \mathbf{y}_m \), and on the principal curvatures and directions of the surface in \( \mathbf{x} \in \Gamma \).

We now have all the ingredients to write out the linear system we need to solve: \( \forall \mathbf{y}_k \in T^\rho_{\varepsilon}, \)
\[ \lambda_1 \rho_1(\mathbf{y}_k) + h^3 \sum_{\mathbf{y}_m \in T^\rho_{\delta}(\mathbf{y}_k) \setminus N_\rho(\mathbf{y}_k)} (\mathbf{K}_{11}(\mathbf{y}_m, \mathbf{y}_k) \rho_1(\mathbf{y}_m) - \mathbf{K}_{12}(\mathbf{y}_m, \mathbf{y}_k) \rho_2(\mathbf{y}_m)) \delta_{\Gamma, \tau}(\mathbf{y}_m) \]
\[ + h^2 \sum_{\mathbf{y}_m \in N_\rho(\mathbf{y}_k)} \left( \omega_{1m}^{(11)} \rho_1(\mathbf{y}_m) - h \omega_{1m}^{(12)} \rho_2(\mathbf{y}_m) \right) \delta_{\Gamma, \tau}(\mathbf{y}_m) = g_1(\mathbf{y}_k), \]
\[ \lambda_2 \rho_2(\mathbf{y}_k) + h^3 \sum_{\mathbf{y}_m \in T^\rho_{\delta}(\mathbf{y}_k) \setminus N_\rho(\mathbf{y}_k)} (\mathbf{K}_{21}(\mathbf{y}_m, \mathbf{y}_k) \rho_1(\mathbf{y}_m) - \mathbf{K}_{22}(\mathbf{y}_m, \mathbf{y}_k) \rho_2(\mathbf{y}_m)) \delta_{\Gamma, \tau}(\mathbf{y}_m) \]
\[ + h^2 \sum_{\mathbf{y}_m \in N_\rho(\mathbf{y}_k)} \left( \omega_{2m}^{(21)} \rho_1(\mathbf{y}_m) - \omega_{2m}^{(22)} \rho_2(\mathbf{y}_m) \right) \delta_{\Gamma, \tau}(\mathbf{y}_m) = g_2(\mathbf{y}_k), \]
where \( \rho_1(\mathbf{y}_k) \approx \bar{\psi}(\mathbf{y}_k) \) and \( \rho_2(\mathbf{y}_k) \approx \bar{\psi}_n(\mathbf{y}_k) \). Corresponding to the target node \( \mathbf{y}_k \) the set \( N_\rho(\mathbf{y}_k) \) contains all the correction nodes, and \( \omega_{ij}^{(m)} = \omega(s_{ij}; \alpha_{mk}, \beta_{mk}), \) \( i, j = 1, 2, \) are the correction weights, dependent on the kernel they correct via the function \( s_{ij} \), and on \( (\alpha_{mk}, \beta_{mk}) \), parameters dependent on \( h, \mathbf{y}_m, \) and \( \mathbf{y}_k \). The details of how the corrections nodes are chosen are presented in Section 2.4, while the definition and computation of the weights is presented in Section 2.4.2.

The contribution of this paper are two quadrature rules. One is a high order, trapezoidal rule-based, quadrature rule for \( J \) via \( J_2 \), effective for smooth surfaces which are globally \( C^2 \). The second is a hybrid rule which combines the previous trapezoidal rule-based with the constant regularization from [22].

### 2.3 Kernels regularization in IBIM - K-reg

The principle of the regularization proposed in [22] is to substitute the kernel \( \mathbf{K} \) with a regularized version \( \tilde{\mathbf{K}} \) around the singularity point, so that the integral over the domain would be as close as possible to the original one. Thus the technique is independent of the quadrature rule. Given a parameter \( \tau > 0 \) and \( \mathbf{x}, \mathbf{y} \in T^\rho_{\varepsilon} \), the regularization of the kernel is
\[ \mathbf{K}_\Gamma(\mathbf{x}, \mathbf{y}) := \begin{cases} \mathbf{K}(\mathbf{x}, \mathbf{y}), & \text{if } |\mathbf{x} - \mathbf{y}|_P \geq \tau, \\ C_{\Gamma, \tau}, & \text{if } |\mathbf{x} - \mathbf{y}|_P < \tau, \end{cases} \]
where \( |\mathbf{x} - \mathbf{y}|_P \) is the distance between the projections of \( \mathbf{x} \) and \( \mathbf{y} \) on the tangent plane at \( P_\Gamma \mathbf{x} \), and \( C_{\Gamma, \tau} \) is a constant dependent on the surface \( \Gamma \) and on the parameter \( \tau \). The constant \( C_{\Gamma, \tau} \) is constructed so that it behaves in the same way as \( \mathbf{K} \) in a neighborhood of \( \mathbf{x} \) dependent on \( \tau \).

Specifically, given \( V(P_\Gamma \mathbf{x}; \tau) \) disc of radius \( \tau \) on the tangent plane of \( \Gamma \) at \( P_\Gamma \mathbf{x} \), it is defined as
\[ C_{\Gamma, \tau} = \frac{1}{V(P_\Gamma \mathbf{x}; \tau)} \int_{V(P_\Gamma \mathbf{x}; \tau)} \mathbf{K}(\mathbf{x}, P_\Gamma \mathbf{z}) d\sigma_{\mathbf{z}}. \]

Then, for the kernels (1.4), the constants are
\[ C_{\Gamma, \tau}^{(11)} = C_{\Gamma, \tau}^{(22)} = C_{\Gamma, \tau}^{(21)} = 0, \quad C_{\Gamma, \tau}^{(12)} = \frac{\exp(-\kappa \tau) - 1 + \kappa \tau}{2\pi \kappa \tau^2}, \]
for \( K_{11}, K_{22}, K_{21}, \) and \( K_{12} \) respectively.

---

1Please note that the weight for the kernel \( K_{12} \) has a different scaling compared to the other three. This is because in order to reach order of accuracy two for such kernel (see (1.6)) the correction weight is zero, so we apply the correction necessary to reach order three.
2.4 The corrected trapezoidal rules in three dimensions - CTR

In this section we present how to construct the corrected trapezoidal rules used in (2.14)-(2.15). The three dimensional quadrature rules will be defined as the sum of integration over different coordinate planes. The two dimensional corrected trapezoidal rule is applied to approximate the integration over each plane. The rule belongs to a class of corrected trapezoidal rules in \( \mathbb{R}^n \) for which convergence results have been proven [8]. The particular selection of the coordinate planes depends on the normal vector of \( \Gamma \) at the target point located on the surface. To make the exposition clear, we will adopt the following convention for this section.

**Notation 2.1.** In the standard basis \((e_1, e_2, e_3)\) of \( \mathbb{R}^3 \) we distinguish between variables in \( \mathbb{R}^2 \) and \( \mathbb{R}^3 \) by using sans serif variables for vectors in \( \mathbb{R}^2 \) and boldface variables with a bar for vectors in \( \mathbb{R}^3 \). For example, \( x \in \mathbb{R}^2 \) and \( \bar{x} \in \mathbb{R}^3 \). Moreover, given a point \( x \) in the standard basis, we denote by its sans serif character its first two components:

\[
\bar{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x \\ x_3 \end{pmatrix}.
\]

We consider a target point, \( \bar{x}^* = (x^*, y^*, z^*) \in \Gamma \), at which the surface normal is \( \bar{n} = (n_1, n_2, n_3) \), \( |\bar{n}| = 1 \). We define the dominant direction \( \bar{e}_i \) of the normal as the one with the largest component:

\[
i = \arg\max_{j=1,2,3} |n_j|, \quad (\bar{e}_1, \bar{e}_2, \bar{e}_3) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.
\]

Let \( Q \) be the unitary matrix describing the change of coordinates such that \( \bar{e}_i \) coincides with the \( z \)-direction. The matrix \( Q \) corresponds to a column permutation of the \( 3 \times 3 \) identity matrix. Let \( q_3 \) be the third row of \( Q \).

**Notation 2.2.** We denote the points in the new coordinate system obtained by the multiplication by \( Q \) using a tilde: given \( x \) expressed in the standard basis,

\[
\tilde{x} := Qx
\]

is its expression in the new basis. Analogously to Notation 2.1, given a point \( \tilde{u} \) in the new coordinate system, we denote by its sans serif character its first two components:

\[
\tilde{u} = \begin{pmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \end{pmatrix} = \begin{pmatrix} \tilde{u} \\ \tilde{u}_3 \end{pmatrix}.
\]

For ease of notation, we define the projection mapping \( \tilde{P}_\Gamma \) after change of coordinates:

\[
\tilde{P}_\Gamma(\tilde{x}) := P_\Gamma(Q^{-1}\tilde{x}) = P_\Gamma(x).
\]

For simplicity, we denote the integrands in (1.2)-(1.3) using the function \( f \),

\[
f(\bar{y}) := K(\bar{x}^*, P_\Gamma(\bar{y}))(\rho(P_\Gamma(\bar{y}))\delta_{\Gamma,x}(\bar{y})), \quad \bar{y} = (x, y, z).
\]

(2.17)

The dependence on \( \bar{x}^* \) is suppressed in this notation, but we shall see how its role fits in the following derivation.

To approximate (1.2)-(1.3) the standard trapezoidal rule is first applied in the dominant direction, which after change of coordinates corresponds to the third component. With the grid points \( t_k = kh \), we get

\[
\int_{\mathbb{R}^3} f(\bar{y})d\bar{y} \approx h \sum_k \int_{\mathbb{R}^2} f(Q^{-1}\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ kh \end{pmatrix})d\tilde{x}d\tilde{y}.
\]

(2.18)

We note that \( f \) is singular in the new coordinate system along the line

\[
\tilde{u}_0(t) = Q\bar{x}^* + \frac{t - q_3\bar{x}^*}{q_3\bar{n}}Q\bar{n}, \quad t \in \mathbb{R}.
\]

(2.19)
parameters respectively, i.e. principal curvatures, principal directions, normal). Moreover, note that the type of singularity for \( s \) plane:

We denote by \( \tilde{u} \) vector recording the first two coordinates of \( \tilde{u} \):

where

is singular at one point.

In the following, for any fixed \( t \) we derive the the first two terms of a series expansion for each of the kernels (1.4) around the singularity. Then a corrected trapezoidal rules for two dimensional integrals in (2.18) are derived based on the expansions.

From the definition of \( \tilde{u}_0(t) \), its third component is \( t \), and we define the points \( \tilde{u}(t) \) as having third component equal to \( t \):

where \( \tilde{u}_0(t) \) denotes the vector containing the first two components of \( \tilde{u}_0(t) \), and \( \tilde{u} \) denotes the vector recording the first two coordinates of \( \tilde{u}(t) \). Then we factorize \( f \), for a fixed \( t \), as

where

Note that the type of singularity for \( s \) depends on the properties of \( \Gamma \) at the target point (such as principal curvatures, principal directions, normal). Moreover, \( s \) depends smoothly on \( t \).

We then use the second order corrected trapezoidal rule \( \mathcal{U}_h^2 \) to compute the integrals on each plane:

We denote by \( \tilde{u}_h(t) \) and \( (\alpha(t), \beta(t)) \) the closest grid node to \( \tilde{u}_0(t) \) and the relative grid shift parameters respectively, i.e.

From the definition in (2.21) we can compute the expansion

The relevant expressions for \( s_k \) are given in Theorem 2.3 below. The second order rule is:

since

By the assumption on \( Q\tilde{n} \), i.e. that the third component of \( Q\tilde{n} \) is the dominant, the normal does not lie on the plane \( xy \). Then for any fixed \( t \)

\[
\int_{\mathbb{R}^2} f \left( Q^{-1} \left( \begin{array}{c} \tilde{u} \\ t \end{array} \right) \right) d\tilde{u} = \int_{\mathbb{R}^2} s(\tilde{u} - \tilde{u}_0(t_k); t_k)v \left( \begin{array}{c} \tilde{u} \\ t_k \end{array} \right) d\tilde{u} \\
\approx \mathcal{U}_h^2 \left[ s(\cdot - \tilde{u}_0(t_k); t_k)v \left( \begin{array}{c} \cdot \\ t_k \end{array} \right) \right].
\]
We define
\[ y_{h,k} := Q^{-1} \left( \begin{array}{c} \bar{u}_h(t_k) \\ t_k \end{array} \right) = Q^{-1} \bar{u}_h(t_k), \]
so the three-dimensional second order method \( V^2_h \), obtained by applying \( U^2_h \) (2.23) plane-by-plane along the dominant direction, is given by

\[ V^2_h[f] := h \sum_{k \in \mathbb{Z}} U^2_h \left[ f \left( Q^{-1} \left( \begin{array}{c} t_k \end{array} \right) \right) \right] \]

(2.24)

\[ = h^3 \sum_{y \in h \mathcal{Z}^2 \setminus \left( \bigcup_{k \in \mathbb{Z}} \{Q^{-1} \bar{u}_h(t_k)\} \right)} f(\bar{y}) + h^2 \sum_{k \in \mathbb{Z}} \omega[| \cdot |^{-1} s_0(\cdot; t_k) ; \alpha(t_k), \beta(t_k)] v \left( \bar{u}_h(t_k) \begin{array}{c} t_k \end{array} \right) \]

(2.25)

\[ = h^3 \sum_{y \in h \mathcal{Z}^2 \setminus \{y_{h,k}\}} f(\bar{y}) + h^2 \sum_{k \in \mathbb{Z}} \omega[| \cdot |^{-1} s_0(\cdot; t_k) ; \alpha(t_k), \beta(t_k)] \rho(\bar{y}_{h,k}) \delta_{T,x}(\bar{y}_{h,k}). \]

### 2.4.1 Expansions of the Poisson-Boltzmann kernels

In this section we will analyze and expand as (2.22) the singular functions defined in (2.21) when \( K \) are the Poisson-Boltzmann kernels (1.5). Specifically we will find the first expansion term \( s_0 \) for all kernels and in addition find \( s_1 \) for the kernel \( K^{12} \) which has a different asymptotic behavior.

\[ s^{(1)}(\bar{u}; t) = \left\{ \frac{\partial G_0}{\partial \bar{n}_y} - \frac{\varepsilon_E}{\varepsilon I} \frac{\partial G_{\kappa}}{\partial \bar{n}_y} \right\} \left( x^*, \bar{P}_T \left( \begin{array}{c} \bar{u} + \bar{u}_0(t) \\ t \end{array} \right) \right), \]

\[ s^{(12)}(\bar{u}; t) = \left\{ G_0 - G_{\kappa} \right\} \left( x^*, \bar{P}_T \left( \begin{array}{c} \bar{u} + \bar{u}_0(t) \\ t \end{array} \right) \right), \]

\[ s^{(21)}(\bar{u}; t) = \left\{ \frac{\partial^2 G_0}{\partial \bar{n}_x \partial \bar{n}_y} - \frac{\partial^2 G_{\kappa}}{\partial \bar{n}_x \partial \bar{n}_y} \right\} \left( x^*, \bar{P}_T \left( \begin{array}{c} \bar{u} + \bar{u}_0(t) \\ t \end{array} \right) \right), \]

\[ s^{(22)}(\bar{u}; t) = \left\{ \frac{\partial G_0}{\partial \bar{n}_x} - \frac{\varepsilon_I}{\varepsilon E} \frac{\partial G_{\kappa}}{\partial \bar{n}_x} \right\} \left( x^*, \bar{P}_T \left( \begin{array}{c} \bar{u} + \bar{u}_0(t) \\ t \end{array} \right) \right). \]

We base the expansion on the local properties of \( \Gamma \) around the target point \( x^* \). Let \( \bar{r}_1, \bar{r}_2, \) and \( \bar{n} \) be the principal directions and outward normal in \( x^* \). Let \( M \) be the diagonal matrix of the principal curvatures, and after change of coordinates let \( A \in \mathbb{R}^{2 \times 2}, c, d \in \mathbb{R}^{2 \times 1}, \) and \( \alpha \in \mathbb{R} \) be the submatrices of the orthogonal change of basis matrix:

\[ M := \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix}, \quad \begin{pmatrix} A & c \\ d^T & \alpha \end{pmatrix} := \begin{pmatrix} -Q\bar{r}_1 & - \\ -Q\bar{r}_2 & - \\ -Q\bar{n} & - \end{pmatrix}. \]

(2.26)

Let \( \eta(t) := d_T(Q^{-1} \bar{u}_0(t)) \), and

\[ D_0(t) := (I - \eta(t)M)^{-1}. \]

We now have all the definitions to express the expansion terms.

**Theorem 2.3.** Let \( x^* \in \Gamma \) be the target point. Suppose that \( Q^{-1} \bar{u}_0(t) \in T_x \). Then, there is an \( r > 0 \), depending on \( t \), such that all the singular functions defined in (2.25) can be written in the form

\[ s^{(ij)}(\bar{u}; t) = \frac{1}{|\bar{u}|} \ell^{(ij)}(\bar{u}) \left( \begin{array}{c} \bar{u} \\ |\bar{u}|t \end{array} \right), \quad |\bar{u}| < r, \quad i,j = 1,2, \]

(2.27)

where \( \ell^{(ij)} \in C^\infty((-r,r) \times S^1) \). Moreover, the functions \( s_0^{(ij)}(\bar{y}; t) : S^1 \times \mathbb{R} \rightarrow \mathbb{R}, i,j = 1,2 \) and
For principal curvatures. If we also use the third derivatives of \( R \) coincide with the standard basis in accuracy, if we improve the local approximation of the surface around the target point \( \bar{s} \), we can approximate and tabulate the weights

\[
\omega[\{s_{ij}(\cdot; t_k); \alpha(t_k), \beta(t_k)\}] =
\]

where

\[
\omega[\{s_{ij}(\cdot); \alpha, \beta\}] = a_0 \omega \left[ |y|^{-1}; \alpha, \beta \right] + \sum_{k=1}^{\infty} \left( a_k \omega \left[ |y|^{-1} \cos(\psi(y)) ; \alpha, \beta \right] + b_k \omega \left[ |y|^{-1} \sin(\psi(y)) ; \alpha, \beta \right] \right).
\]

We can then approximate and tabulate the weights \( \omega[\{s_{ij}(\cdot); \alpha, \beta\}] \) in the following way. We fix a stencil of parameters \( \{(\alpha_m, \beta_n)\}_{m,n} \) around \( (\alpha, \beta) \) and basis functions \( \{c_{m,n}(\alpha, \beta)\}_{m,n} \) such that we can approximate a function \( f : \mathbb{R}^2 \to \mathbb{R} \) in \( (\alpha, \beta) \) as

\[
f(\alpha, \beta) \approx \sum_{m,n} c_{m,n}(\alpha, \beta) f(\alpha_m, \beta_n).
\]

We let \( N \) be the number of Fourier modes used to approximate the weights. Then, given \( \ell \), we first find the \( 2N + 1 \) coefficients \( a_0, \{a_k, b_k\}_{k=1}^{N} \) by using the Fast Fourier Transform. Then,

\[
\omega[\{s_{ij}(\cdot); \alpha_m, \beta_n\}] \approx a_0 \omega \left[ |y|^{-1}; \alpha_m, \beta_n \right] + \sum_{k=1}^{N} \left( a_k \omega \left[ |y|^{-1} \cos(\psi(y)) ; \alpha_m, \beta_n \right] + b_k \omega \left[ |y|^{-1} \sin(\psi(y)) ; \alpha_m, \beta_n \right] \right),
\]

As previously mentioned, the kernel \( K_{21} \) is the bottleneck for the order of accuracy. We can find more expansion terms for the function \( s^{(ij)} \) in (2.27), and consequently increase the order of accuracy, if we improve the local approximation of the surface around the target point \( \bar{x}^* \).

We rotate and translate \( \Gamma \) so that \( \bar{x}^* \) is in the origin and the principal directions and the normal coincide with the standard basis in \( \mathbb{R}^3 \). Then \( \Gamma \) can be represented as the graph of a function \( F \). In our current approach we are using the (pure) second derivatives of the function \( F \), i.e. the principal curvatures. If we also use the third derivatives of \( F \), we can write the expansion terms

\[
s^{(11)}(\bar{s}; t), \quad s^{(22)}(\bar{s}; t), \quad s^{(12)}(\bar{s}; t),
\]

which we can use to build a third order accurate corrected trapezoidal rule, as we showed in [10]. However, we cannot write one corresponding to \( K_{21} \), i.e. \( s^{(21)}(\bar{s}; t) \). To find such term, we need not only the third derivatives of \( F \), but also the fourth derivatives.

### 2.4.2 Approximation and tabulation of the weights

Given the functions (2.28) we want to compute the weights \( \omega[\{s_{ij}(\cdot; t_k); \alpha(t_k), \beta(t_k)\}] \). The functions in the expansion (2.22) are of the kind

\[
\frac{1}{|\ell|} \ell \left( \frac{y}{|y|} \right), \quad y \in \mathbb{R}^2,
\]

for \( s^{(ij)} \) where \( i, j = 1, 2 \) or, in the case of \( s^{(12)} \), simply of the kind \( \ell \left( \frac{y}{|y|} \right) \). Fixed \( (\alpha, \beta) \), we write \( \ell \) using its Fourier series:

\[
y = |y| \left( \cos(\psi(y)), \sin(\psi(y)) \right), \quad \ell \left( \frac{y}{|y|} \right) = a_0 + \sum_{k=1}^{\infty} \left( a_k \cos(k \psi(y)) + b_k \sin(k \psi(y)) \right),
\]

where \( \{a_j\}_{j=0}^{\infty} \) and \( \{b_j\}_{j=1}^{\infty} \) are the Fourier coefficients of \( \ell \). By linearity of the weights with respect to \( \ell \), we can write them as

\[
\omega[\{s_{ij}(\cdot); \alpha, \beta\}] = a_0 \omega \left[ |y|^{-1}; \alpha, \beta \right] + \sum_{k=1}^{\infty} \left( a_k \omega \left[ |y|^{-1} \cos(k \psi(y)) ; \alpha, \beta \right] + b_k \omega \left[ |y|^{-1} \sin(k \psi(y)) ; \alpha, \beta \right] \right).
\]

We can then approximate and tabulate the weights \( \omega[\{s_{ij}(\cdot); \alpha, \beta\}] \) in the following way. We fix a stencil of parameters \( \{(\alpha_m, \beta_n)\}_{m,n} \) around \( (\alpha, \beta) \) and basis functions \( \{c_{m,n}(\alpha, \beta)\}_{m,n} \) such that we can approximate a function \( f : \mathbb{R}^2 \to \mathbb{R} \) in \( (\alpha, \beta) \) as

\[
f(\alpha, \beta) \approx \sum_{m,n} c_{m,n}(\alpha, \beta) f(\alpha_m, \beta_n).
\]

We let \( N \) be the number of Fourier modes used to approximate the weights. Then, given \( \ell \), we first find the \( 2N + 1 \) coefficients \( a_0, \{a_k, b_k\}_{k=1}^{N} \) by using the Fast Fourier Transform. Then,

\[
\omega[\{s_{ij}(\cdot); \alpha_m, \beta_n\}] \approx a_0 \omega \left[ |y|^{-1}; \alpha_m, \beta_n \right] + \sum_{k=1}^{N} \left( a_k \omega \left[ |y|^{-1} \cos(k \psi(y)) ; \alpha_m, \beta_n \right] + b_k \omega \left[ |y|^{-1} \sin(k \psi(y)) ; \alpha_m, \beta_n \right] \right),
\]

and
and we can approximate the weight for \((\alpha, \beta)\) via
\[
\omega[s^{(ij)}_q; \alpha, \beta] \approx \sum_{m,n} c_{m,n}(\alpha, \beta) \omega[s^{(ij)}_0; \alpha_m, \beta_n].
\]

We consequently need to compute and store the weights for the following constant and trigonometric functions,
\[
\begin{align*}
\omega[|y|^{-1}; \alpha_m, \beta_n] \\
\omega[|y|^{-1}\cos(k\psi(y)); \alpha_m, \beta_n] \\
\omega[|y|^{-1}\sin(k\psi(y)); \alpha_m, \beta_n]
\end{align*}
\]

\(k = 1, \ldots, N,\) and all \(m, n\) in the stencil for \((\alpha, \beta)\).

For \(s^{(12)}_1\), the function is constant with respect to \(|y|\) and \(y/|y|\), so the only weights needed are \(\omega[1; \alpha_m, \beta_n]\). For ease of notation, let’s write \(s_q(y) := |y|^{q-1}(y/|y|)\), where \(q \in \{0, 1\}\). The weights \(\omega[s_q; \alpha_m, \beta_n]\) are formally the limits
\[
\omega[s_q; \alpha, \beta] := \lim_{h \to 0^+} \omega_h[s_q; \alpha, \beta],
\]
where
\[
\omega_h[s_q; \alpha, \beta] := \frac{1}{h^{q+1}} \int_{B^2} s_q(x) g(x) dx - T_h^q \left[ s_q(\cdot - (\alpha, \beta)h) g(\cdot - (\alpha, \beta)h) \right] / g(- (\alpha, \beta)h).
\]

The function \(g\) is chosen to be \(g \in C^\infty_c(\mathbb{R}^2)\) and radially symmetric, with \(g(0) = 1\).

**Remark 2.4.** We approximate the limit (2.29) by \(\omega_{1, h^*}\), where
\[
h^* := 2^{-n}, \quad n := \arg \min_{j=1,2,3,\ldots} \{|\omega_{1,2-j} - \omega_{1,2-j-1}| \leq \text{Tol}\}.
\]

In the simulations presented in Section 2.6, to compute \(\omega\), we use \(\text{Tol} = 10^{-8}\).

### 2.5 Overview of the algorithm

We now have all the tools to build the system (2.14)-(2.15). Let \(\{y_k\}_{k=1}^M\) be the discretization nodes inside the tubular neighborhood. We write the system in matrix form as
\[
\Lambda p + h^4KWp + h^2\Omega Wp = g
\]
where \(p\) contains both \(\tilde{\psi}(y_k)\) and \(\tilde{\psi}_n(y_k)\), and
\[
\Lambda := \begin{pmatrix} \lambda_1 I & 0 \\ 0 & \lambda_2 I \end{pmatrix}.
\]

The matrix \(W\) is a diagonal matrix defined by the weights \(\delta_{r,e}(y_k)\), and the matrices \(K\) and \(\Omega\) represent the dense matrix of the kernel evaluations \(K(P_r y_k, y_m)\), \(k, m = 1, \ldots, M\) except in the corrected nodes, and the sparse matrix of the weights for the corrected nodes respectively. In Figure 1 we see the complementary structure of the matrices \(K\) and \(\Omega\), where the missing elements in \(K\) correspond to the corrected terms in \(\Omega\).

We solve this system by a standard GMRES solver. In the GMRES algorithm, we use the black-box fast multipole method BBFMM (see [5]) to accelerate the multiplication of the operator \(K\) to any vector. The codes are available on GitHub\(^2\).

Compared to the K-reg method, the CTR2 has a similar computational cost. The weights require interpolation of the tabulated values, which is negligible. For the second order method described here, CTR2, only one node per plane needs correction, which is similar to K-reg is the parameter \(\tau = h\) or \(2h\). The corrected nodes for CTR2 and the regularized nodes for K-reg lead

\(^2\)https://github.com/lowrank/ibim-levelset
Figure 1: Sparsity pattern of the matrix $\Omega$ and complementary form of the matrix $K$. The surface is the SES (see Section 3.1) for the union of two intersecting spheres of different radii. The tubular neighborhood is of width $\varepsilon = 2h$, which means that in the best possible case 5 nodes are corrected for a given target node. The weights present in $\Omega$ do not change the removed values from $K$ and correct them according to the CTR2 rule. Left: non-zero elements of $\Omega$, which has approximately 4 non-zero elements for each row. Right: the zero elements of the dense matrix $K$ corresponding to the ones removed in order to apply the correction.

The CTR2 needs to identify all the closest nodes along the singular line and interpolate the weights. This is done once in precomputation, as the matrix used in the matrix-vector multiplication does not change along the GMRES iterations.

In the following Section, we will present the number of GMRES iterations needed to solve the system for the methods used, and see that CTR2 has a similar condition number compared to K-reg, needing a small number of iterations to converge to the desired accuracy.

### 2.6 Numerical tests

In this section we present two numerical studies on the order of accuracy of our method (CTR2). We compare the results to those computed with the IBIM regularization (K-reg) method described in Section 2.3.

The surfaces used in our tests include a sphere with radius $r = 10$ and a torus with radii $R_1 = 1$ and $R_2 = 1/2$, both centered in the origin. The torus is rotated along the $x$, $y$, and $z$-axes by the following angles respectively:

$$a = 1.99487, \quad b = 2.54097947651017, \quad c = 4.219760487439292.$$

We assess our method by solving numerically (1.2)-(1.3) on the grid $h\mathbb{Z}^3$ with right hand side $g$ built using a single charge at $0$ with charge $q$. The constants $\varepsilon_I$ and $\varepsilon_E$ are fixed at:

$$\varepsilon_I = 1.0, \quad \varepsilon_E = 80.0.$$

Then we find the solutions $\psi$ and $\psi_n = \partial\psi/\partial n$. Then we compute the integral

$$\psi_{\tau z n}(z) := \int_{\Gamma} \left\{ \left( \frac{\varepsilon_E}{\varepsilon_I} \frac{\partial G_\kappa}{\partial n_y}(z, y) - \frac{\partial G_0}{\partial n_y}(z, y) \right) \psi(y) + (G_0(z, y) - G_\kappa(z, y)) \frac{\partial\psi}{\partial n}(y) \right\} d\sigma_y, \quad (2.31)$$

Then we find the solutions $\psi$ and $\psi_n = \partial\psi/\partial n$. Then we compute the integral

$$\psi_{\tau z n}(z) := \int_{\Gamma} \left\{ \left( \frac{\varepsilon_E}{\varepsilon_I} \frac{\partial G_\kappa}{\partial n_y}(z, y) - \frac{\partial G_0}{\partial n_y}(z, y) \right) \psi(y) + (G_0(z, y) - G_\kappa(z, y)) \frac{\partial\psi}{\partial n}(y) \right\} d\sigma_y, \quad (2.31)$$
where \( z \) is a point in space not belonging to the surface. The integral is computed in the IBIM framework (2.6) using the standard trapezoidal rule for

\[
f(y) := \left( \frac{\varepsilon E}{\varepsilon I} \frac{\partial G_\kappa}{\partial n_y} (z, y) - \frac{\partial G_0}{\partial n_y} (z, y) \right) \psi(y) + (G_0(z, y) - G_\kappa(z, y)) \frac{\partial \psi}{\partial n}(y),
\]

and the averaging kernel \( \phi \) (2.7) used is

\[
\phi(x) = \begin{cases} 
\frac{1}{2} (1 + \cos(\pi x)), & \text{if } |x| \leq 1, \\
0, & \text{otherwise.}
\end{cases}
\]

We use the standard trapezoidal rule because \( z \) does not lie close to the surface, so the integrand \( f \) is smooth.

In Figure 2 we compare the convergence of the error in \( \psi_{\text{rxn}}(0) \) for the sphere, for K-reg and CTR2 and different widths of the tubular neighborhood \( T_r \). The exact values \( \psi^* \) and \( \psi^*_n \) of the solution on the sphere are:

\[
\psi^*(x) = \begin{cases} 
\frac{q}{4\pi \varepsilon E |x|} + \frac{q}{4\pi r} \left( \frac{1}{\varepsilon E (1 + \kappa r)} - \frac{1}{\varepsilon I} \right), & |x| \leq r, \\
\frac{q}{4\pi \varepsilon E (1 + \kappa r)} |x|, & |x| > r.
\end{cases}
\]

\[
\psi^*_n(x) = \begin{cases} 
\frac{q}{4\pi \varepsilon I |x|^2} - \frac{k q e^{-\kappa|x|/r}}{4\pi \varepsilon E (1 + \kappa r)} - \frac{k q e^{-\kappa|x|/r}}{4\pi \varepsilon E (1 + \kappa r)} |x|, & |x| \leq r, \\
\frac{k q e^{-\kappa|x|/r}}{4\pi \varepsilon E (1 + \kappa r)} |x|, & |x| > r.
\end{cases}
\]

and the value of \( \psi_{\text{rxn}}(0) \) is

\[
\frac{q^2}{4\pi r} \left( \frac{1}{\varepsilon E (1 + \kappa r)} - \frac{1}{\varepsilon I} \right).
\]

The regularization parameter used for K-reg is \( \tau = 2h \). When \( \varepsilon \) is constant with respect to \( h \), CTR2 exhibits an order of accuracy slightly greater than 2, while K-reg only has order 1. In the case \( \varepsilon = 2h \), the order of accuracy of CTR2 decreases to approximately 1 because \( \delta_{\text{r},r} \) is not well resolved by the grid. Then the order of accuracy is the same as K-reg; however the error constant for CTR2 is smaller by a factor of at least 10 compared to K-reg.

Table 1: Torus test, with \( \varepsilon = 2h \). We compute the potentials \( \psi \) and \( \psi_n \) numerically using K-reg and CTR2, and use them to compute \( \psi_{\text{rxn}}(0) \) (2.31). We use two different approximations of the Jacobian \( J_\tau \). For K-reg, \( J_\tau \approx 1 \), and for CTR2, \( J_\tau \) is approximated to second order in \( h \) by \( J_\tau \approx J_h \). For the surface area \( A \) we directly tabulate the relative error.

| grid size | \( J_\tau \approx 1 \) | \( \psi_{\text{rxn}}(0) \), K-reg | diff | \( J_\tau \approx J_h \) | \( \psi_{\text{rxn}}(0) \), CTR2 | diff |
|-----------|----------------|-----------------|-----|----------------|----------------|-----|
| 103       | 9.483e-6       | 2.213e+2        | 5.6e-1 | 4.498e-6       | 2.2354e+2      | 4.9e-2 |
| 128       | 9.280e-6       | 2.219e+2        | 9.8e-1 | 9.525e-7       | 2.2353e+2      | 1.3e-2 |
| 256       | 2.283e-5       | 2.229e+2        | 3.3e-1 | 2.492e-6       | 2.2352e+2      | 2.5e-3 |
| 512       | 1.783e-6       | 2.232e+2        | 3.3e-1 | 2.492e-6       | 2.2352e+2      | 2.5e-3 |

For the torus we compiled the computed values in Table 1. In the computation of \( \psi_{\text{rxn}}(0) \), K-reg approximates the Jacobian in the IBIM integral (2.6) as constant 1, while CTR2 approximates it using a second order approximation in \( h \), \( J_\tau \approx J_h \). According to \[14\], if \( \phi \) in (2.7) is an even function, then the terms corresponding to the odd powers of the distance in the Jacobian, \( J_\tau \), will average out analytically. Hence, by approximating \( J_\tau \) by 1, one may expect an analytically error of \( \mathcal{O}(\varepsilon^2) \). This fact was exploited for convenience in K-reg. However, when \( \varepsilon \) is only a small constant multiple of \( h \), discrete errors will dominate (see [2]). In CTR2, since curvature information is
Figure 2: Convergence results for a sphere. The methods K-reg and CTR2 of order two are used to discretize the integrals in (1.2)-(1.3) and solve the corresponding system. Then we evaluate $\psi_{\text{rxn}}(x_0)$ in $x_0$ center of the sphere. For the sphere we have analytic values for $\psi_{\text{rxn}}(x_0)$. We plot the relative error for CTR2 (left) and K-reg (right).

required for computing the quadrature weights, it is natural to use a second order approximation of $J_\Gamma$. To gauge the influence of these approximations, we also compute the surface area $A$ by using formula (2.6) with $f \equiv 1$ and the same two ways of approximating $J_\Gamma$. We then compute the relative surface area error, listed in Table 1.

3 Computing electrostatic potential of macromolecules

In this section, we use the IBIM formulation to solve the linearized Poisson-Boltzmann equation and compute the electrostatic potential and polarization energy for solvent-molecule interfaces. The solvent-molecule interface represents the interface separating the solvent fluid particles and the macromolecule. The boundary integral equations are defined on such interface. Such surfaces are naturally complex and difficult to parametrize. Consequently it is difficult to apply standard BIM with an explicit parametrization for these applications. In the following section we briefly review two different approaches for defining the solvent-molecule interface. For either approach the IBIM approach can be conveniently applied. Then the IBIM equations are discretized using CTR. We compare the method with the previous regularization approach K-reg, and introduce a hybrid corrected/regularized method HYB-IBIM to deal with surfaces which are piecewise smooth.

3.1 Solvent-molecule interface

The solvent-molecule interface, $\Gamma$, can be defined in different ways. Classically, $\Gamma$ is approximately by unions of predefined shapes, which leads to less accurate approximations. Nevertheless, one can obtain qualitatively correct information when the errors are averaged over large molecules [6, §22.1.2.1.2].

The van der Waals (VDW) molecular surface is the simplest interface to define: each atom in the protein is approximated by a solid sphere with the van der Waals radius. The union of the regions enclosed by the spheres is the VDW surface. A VDW surface typically has kinks are is only $C^0$.

The solvent-accessible surface (SAS) of a molecule is defined by "rolling" a small sphere around the VDW surface. A small region near the kink in the VDW surface are replaced by a portion of this sphere, and the the resulting SAS is $C^1$ and piece-wise $C^2$. From the SAS one can define the solvent-excluding surface (SES) by removing any cavity enclosed by the molecule, and consequently
inaccessible by the solvent. The radius of the probe is usually set to the radius of the water molecule. The accessibility of solvent actually depends on its three-dimensional orientation, but this dependence is ignored for simplicity because the radius of the oxygen atom is much larger than the hydrogen (see [6, §22.1.2.1.2]). The resulting SES is formally \( C^1 \).

The SES can be quite difficult to find analytically, so we used the numerical approach described in [22]. The information about the SES is in the form of a signed distance function on a uniform Cartesian grid. The steps are summarized here:

1. An initial level set function is defined.
2. The level set function is evolved using an “inward” eikonal flow.
3. The internal cavities are removed.
4. A reinitialization procedure is applied to the level set function, which will give as a result the sought signed distance function to the SES.

The physical phenomena which control the dynamics of the biomolecular systems greatly impact the solvent-molecule interface. Consequently in biomolecular applications, for improving accuracy to the underlying physics, the solvent-molecule interface itself can be an unknown to solve for. More recently methods have been developed which take the relevant dynamics into account, and find the solvent-molecule interface as the minimizer of an energy functional. One such method is the Variational Implicit Solvation Model (VISM).

Figure 3: Solvent-molecule interfaces of p53-MDM2 generated in two different ways. On the left the SES is generated using the code from [22]. The center and right figures show the smooth solvent-molecule interfaces generated via VISM. The two different surfaces are generated using two different initial configurations of the complex biomolecular system p53-MDM2 (PDB ID 1YCR) [15] from the Protein Data Bank (PDB), as detailed in [21]. The different initial configurations can lead to different minimizers to the free-energy functional because of its non-convex nature. This is detailed in the original algorithm paper of VISM [19]. The surfaces are plotted using the signed distance function values. Left: SES surface. Center: VISM “loose” surface, generated with an initial configuration further from the atoms. Right: VISM “tight” surface, generated with an initial configuration closer to the atoms.

In VISM, the solvent-molecule interface is found by iteratively minimizing the free-energy functional over a surface, as for example seen in [19]:

\[
\mathcal{Z}[\Gamma] = \mathcal{G}_{\text{geom}}[\Gamma] + \mathcal{G}_{\text{vdW}}[\Gamma] + \mathcal{G}_{\text{elec}}[\Gamma],
\]

where the electrostatic interaction \( \mathcal{G}_{\text{elec}}[\Gamma] \) is approximated using the Coulomb-field approximation. Typically, this energy is minimized by computing a sequence of minimizing surfaces, \( \Gamma_k \), \( k = 0, 1, 2, \ldots \). For each \( k \) an electrostatic problem is solved to evaluate \( \mathcal{G}_{\text{elec}}[\Gamma_k] \). While other formulations of the electrostatic energy can be used, e.g. the Coulomb-field approximation (CFA) (see e.g. [19, 20]) one of the most accurate is provided by the Poisson-Boltzmann theory (see e.g. [16, 23, 27]).

Different methods lead to solvent-molecule interfaces of different smoothness:
The SAS is of smoothness $C^0$ or $C^1$ depending on which set of points is used. In the case of the SAS defined as the set of center points of the probe as it rolls the surface can have kinks and consequently be of regularity at most $C^0$. In the case of the SAS defined as the set of points tangent to the model the surface is of regularity $C^1$ because the curvatures will be discontinuous when jumping from an atom to the probe.

The SES is formally of the same smoothness of the SAS taken with points tangent to the model.

The surface generated with VISM is smoother due to the presence of the surface tension term in $G_{geom}[\Gamma]$, one of the terms in the energy to be minimized. We point out that IBIM requires that the solvent-molecule interface to be at least $C^{1,\alpha}$ for some $\alpha > 0$. This way, one can be sure that there exists a workable tubular neighborhood. The second order CTR, CTR2, formally requires the surfaces to be $C^2$ almost everywhere on the surface, since principal curvatures and directions on the surface are needed for computing the quadrature weights. While SES surfaces formally do not present any problem on the analytical level, the approximation of these second order geometrical quantities on an SES, near the discontinuities of these surface quantities, may introduce non-negligible errors to the integration and to the discretized linear system. In the following subsections, we propose a hybrid method for solving the linearized Poisson-Boltzmann equation on the SES. We will show that the hybrid method retains second order of accuracy if, for a piecewise smooth surface, the number of points where the curvatures are undefined scales linearly with $h$.

### 3.2 Hybrid corrected/regularized quadrature

We classify the nodes inside the tubular neighborhood into nodes which are “bad” and “good”,

$$\mathcal{M} := \{x_m\}_{m=1}^M = h\mathbb{Z}^3 \cap T_e = \mathcal{M}_G \cup \mathcal{M}_B,$$

depending on whether the stencil to compute the second derivatives of the signed distance function is negatively impacted by the curvature discontinuity. By taking the corrected trapezoidal rule for the “good” points where the curvatures are relatively accurate, and the regularized quadrature rule for the “bad” points, we expect a second order accuracy in the grid size $h$ for the good points and only first order for the bad points. The overall order of accuracy of the hybrid approach will then depend on the percentage of the bad points.

For good and bad target points, the quadrature rules used to discretize the equations (1.2)-(1.3) are CTR2 and K-reg respectively. Then the system (2.30), after reordering so that the nodes are grouped together, becomes

$$\begin{pmatrix}
\Lambda|\mathcal{M}_G| + \frac{h^3}{3}K_B W_B + \frac{h^2}{2} \Omega_{K_{reg}} W_B
\Lambda|\mathcal{M}_B| + \frac{h^3}{3}K_G W_G + \frac{h^2}{2} \Omega_{CTR} W_G
\end{pmatrix}
\begin{pmatrix}
p_B
p_G
\end{pmatrix}
= \begin{pmatrix}
g_B
g_G
\end{pmatrix}.
$$

(3.1)

The vector $p_B$ groups the values of $\psi$ and $\psi_n$ in the “bad” points, and analogously for $p_G$. Depending on whether the target point is in $\mathcal{M}_G$ or $\mathcal{M}_B$, the nodes which are corrected or regularized may differ. Consequently we write the subscript $G$ and $B$ respectively.

The scheme will have order of accuracy $O(h^{3/2})$ if the number of bad points scales at most linearly in $h^{-1}$. We see this by rewriting the system (3.1) in the simplified representation

$$(\Lambda + h^3 A)\psi = b$$

where $A$ represents the quadrature discretization matrix, $\psi$ represents the whole solution vector and $b$ the right hand side.

Given two nodes $x_i \in \mathcal{M}_G$ and $x_j \in \mathcal{M}_B$ in the good and bad sets respectively, then we can formally estimate the error of the quadrature rule as

$$h^3 A(i,:)\psi = S[\psi](x_i) + O(h^2),$$

$$h^3 A(j,:)\psi = S[\psi](x_j) + O(h),$$

where $S[\psi]$ represents the signed distance function of $\psi$ at the point $x_i$ or $x_j$. The error is bounded by $O(h^2)$ for the good points and $O(h)$ for the bad points, leading to the overall second order accuracy of the hybrid method.
where
\[ S[\hat{\psi}](x_i) = \int_{T} K(P_{T}x_i, y)\hat{\psi}(y)\delta_{\Gamma,e}(y)dy \]
is the linear integral operator we are approximating. We call \( \hat{\psi} \) the solution to the system
\[ \lambda \hat{\psi}_i + S[\hat{\psi}](x_i) + d_i(h) = b, \quad i = 1, \ldots, M, \]
where
\[ d(h) := (h, h, \ldots, h, h^2, h^2, \ldots, h^2)_{\mathcal{M}_B}, \]
and \( \psi \) is the solution to the system
\[ \lambda \psi_i + S[\psi](x_i) = b, \quad i = 1, \ldots, M. \]
Then the difference between the two solutions is:
\[ \psi - \hat{\psi} = (\lambda I + S)^{-1}d(h). \]
In the case \( M \sim h^{-2} \) and \( |\mathcal{M}_B| \sim h^{-1} \), then the dependency of the mean \( \ell_2 \)-error of \( \psi - \hat{\psi} \) on \( h \) is
\[ |\psi - \hat{\psi}| \leq |(\lambda I + S)^{-1}||D(h)| = C \left( \sum_{i=1}^{\mathcal{M}_B} h^2 + \sum_{i=1}^{\mathcal{M}_G} h^4 \right) = C \sqrt{h^{-1}h^2 + h^{-2}h^4} \sim O(\sqrt{h}) \]
\[ \Rightarrow \frac{|\psi - \hat{\psi}|}{\sqrt{M}} \sim O(h^{3/2}). \]
Then the averaged order of accuracy for the hybrid rule is formally 3/2, still larger than 1, if the number of bad points scales linearly with \( h^{-1} \).

We call this hybrid method HYB-IBIM, and remark that the good and bad points are also used to identify which approximation of the Jacobian \( J_{\Gamma} \). In the good points we use the second order approximation \( J_{\Gamma} \approx J_{h} \), while in the bad points we use the approximation \( J_{\Gamma} \approx 1 \). We indicate this approximation of the Jacobian (2.8) by
\[ J_{h}^{(H)}(y) := \begin{cases} 1, & \text{if } y \in \mathcal{M}_B, \\ J_{h}(y), & \text{if } y \in \mathcal{M}_G. \end{cases} \tag{3.2} \]

3.3 Numerical tests
In the following, we assess our methods by comparing the computed values of surface area and the polarization energy (3.3). The surface area, \( A \), is computed by the standard trapezoidal rule for the integral defined on the right-hand-side (2.6) with \( \rho \) constant equal to 1 and \( \phi \) as in Section 2.6. The polarization energy is defined by the formula:
\[ G_{\text{pol}} := \frac{1}{2} \sum_{k=1}^{N_c} q_k \psi_{\text{rxn}}(z_k), \tag{3.3} \]
\[ \psi_{\text{rxn}}(z) := \int_{\Gamma} \left\{ \left( \frac{\varepsilon E}{\varepsilon I} \frac{\partial G_0}{\partial n_{y}}(z, y) - \frac{\partial G_{\phi}}{\partial n_{y}}(z, y) \right) \psi(y) + \left( G_{0}(z, y) - G_{\phi}(z, y) \right) \frac{\partial \psi}{\partial n}(y) \right\} d\sigma_{y}, \]
where \( \psi \) and \( \partial \psi/\partial n \) are computed by solving (2.14) and (2.15). The surface integral (3.3) is, again, computed by second order CTR, CTR2, applied to the associated IBIM formulation. Because \( z_k \) are the centers of the atoms, they do not fall onto the surface and the CTR2 just becomes the standard trapezoidal rule.
3.3.1 Smooth interfaces

In this section, we apply CTR2 to solve the linearized Poisson-Boltzmann equation on smooth surfaces. The tests run in Section 2.6 include a sphere, which corresponds to a single ion. The value (2.31) corresponds to the polarization energy (up to a constant), and we presented convergence results in Figure 2.

Then we apply CTR2 to solve systems derived from two smooth solvent-molecule interfaces for the protein p53-MDM2 [21]. The interfaces are constructed by the LS-VISM code developed in [24] using two different starting configurations (corresponding to what is called a "loose" and a "tight interface). Because of the complexity of the problem and the protein, we do not have analytical solutions.

The signed distance functions to the two surfaces are computed on the Cartesian grid in $[-31.7945, 31.7945]^3$, accurate within the absolute distance of $1.24197$ to each interface.

We see in Figure 3 that the tight surface has two closely positioned connected components. In physical units we can estimate the bound on the reach to be approximately $2.5$. The stencil of the finite differences used to approximate the curvatures to second order in $h$ is 5 grid nodes in each coordinate direction. This means that we the $513^3$ uniform Cartesian grid ($h \approx 0.12$) can provide roughly 10 points to discretize the tight spacing between the two component, supporting a tubular neighborhood of width $\varepsilon = 2h$.

In Tables 2 and 3 we tabulate the surface area and solve the system with right hand side computed using a single fictitious ion at the origin, which falls inside the loose molecule but outside the tight one. Then we evaluate $\psi_{\text{rxn}} \ (2.31)$ at this point, and its sign depends whether the evaluation point lies inside or outside of the surface. The results are obtained for $\varepsilon = 2h$ and $5h$, and we use the difference between them to get a rough estimate of the accuracy of the values.

In a different test, we solve the system for the tight smooth surface by using as centers the centers of the atoms, and compute the polarization energy $G_{\text{pol}} \ (3.3)$. We use both K-reg and CTR2, and see that the system is well-conditioned by comparing the number of GMRES iterations needed. The potential obtained is plotted in Figure 4.

Table 2: Smooth “loose” solvent-molecule interface results. We solve the system on the surface presented on the left in Figure 3, using K-reg and CTR2. Then we evaluate the function $\psi_{\text{rxn}} \ (2.31)$ in the origin to test the accuracy of the methods. Concurrently we test the area approximation with the two different Jacobian approximations used in the two different methods.

| grid size | width $\varepsilon$ | d.o.f. | $A, J_T \approx 1$ | $\psi_{\text{rxn}}(0)$, K-reg | GMRES |
|-----------|---------------------|-------|--------------------|---------------------------------|-------|
| $513^3$   | $2h$                | 1468195 | 5661.51           | -1.6677e+1                      | 11    |
| $513^3$   | $5h$                | 3671757 | 5662.14           | -1.6671e+1                      |       |

Table 3: Smooth “tight” solvent-molecule interface results, analogous to the ones presented in Table 2. The surface can be seen plotted on the right in Figure 3.

| grid size | width $\varepsilon$ | d.o.f. | $A, J_T \approx 1$ | $\psi_{\text{rxn}}(0)$, K-reg | GMRES |
|-----------|---------------------|-------|--------------------|---------------------------------|-------|
| $513^3$   | $2h$                | 1521649 | 5866.30           | 2.2840e+2                       | 12    |
| $513^3$   | $5h$                | 3805197 | 5867.53           | 2.2915e+2                       |       |

17
Figure 4: Electrostatic potential over the “tight” solvent-molecule interface. The electrostatic potential and its normal derivative $\psi$ and $\psi_n$ are approximated numerically using K-reg and CTR2, with tubular neighborhood of width $\varepsilon = 2h$ and Cartesian grid of size $513^3$. The system is inverted in 13 iterations for K-reg and 15 for CTR2. Then we compute the polarization energy $G_{pol}$ and obtain the following values: $-1.4534e+3$ for K-reg and $-1.4478e+3$ for CTR2. The electrostatic potential plotted is computed using CTR2.

3.3.2 Solvent excluded surfaces

The next tests involve the hybrid method, HYB-IBIM, for solvent excluded surfaces (SES), which are globally $C^1$ and piecewise $C^2$.

First we consider a three-sphere model shown in Figure 5. The SES is generated using the method in [22]. In this implementation, “bad points” are classified if the curvatures at that point are away from the true ones. We numerically verify that the number of bad points (ref. Sec 3.2) $|\mathcal{M}_B|$ scales linearly in $h^{-1}$ if $\varepsilon \sim h$, as presented in Figure 5.

Next, with Table 4, we present a numerical convergence study using the three available methods: K-reg, CTR2, and HYB-IBIM. We compute the difference of consecutive values of $\psi_{rnx_0}(0)$ to estimate the accuracy, and see that the CTR2 and HYB methods appear to have higher accuracy than K-reg. In this study, it seems that the difference between CTR2 and HYB is minimal.

Finally, we apply the hybrid method also on a solvent-molecule interface, we take the protein 1YCR and generate the SES using the method from [22]. We run a convergence test on it, seen in Table 5. The electrostatic potential computed with the HYB-IBIM is also plotted for $513^3$ in Figure 6.

4 Conclusions

In this paper, we present an accurate corrected trapezoidal-rule based the numerical solution of the linearized Poisson-Boltzmann equation. Using the boundary integral formulation developed in Juffer et al. [11] we solve the boundary integral equations using a non-parametric approach. In theory [8] it is possible to derive very high order accurate quadratures for corresponding boundary integral equations. However, high order accuracy requires higher order accurate approximations to the surface’s geometry. We identify the kernel $K_{21}$ as the bottleneck for the order of accuracy attainable. Given a second order approximation of the surface locally around the target points, the proposed corrected trapezoidal rule can reach third order of accuracy for integrating all the kernels except $K_{21}$. To obtain a third order approximation for integrating $K_{21}$, a fourth order approximation of the surface geometry is needed; see Section 2.4.1.

We first tested the method on a single sphere/ion and checked the order of accuracy of the
Figure 5: Top: Illustration of the SES for a three-sphere model. Bottom left: $M$ number of nodes interior to the tubular neighborhood, i.e. degrees of freedom, as function of $N$ discretization along a single direction. For $\varepsilon$ linear in $h$, the number of d.o.f. scales as $N^2 \sim h^{-2}$. Bottom right: number $|M_B|$ of “bad” nodes compared with the total number of nodes scales linearly in $h^{-1}$.

Figure 6: Electrostatic potential over the SES of the protein 1YCR, computed using HYD-IBIM.

method. Moreover we assessed that the error constant when using a width of $\varepsilon$ linear in $h$ is much smaller than the regularization method K-reg.

Then we studied the application of the Poisson-Boltzmann equation to compute the electrostatic potential and polarization energy of macromolecules immersed in a solvent. The smoothness of the solvent-molecule interface depends on how it is defined. We presented polarization energies for the smooth interfaces, and then tested a hybrid corrected/regularized method for piecewise $C^2$ surfaces.
Table 4: Three spheres test, with width $\varepsilon = 2h$. We solve the system on the surface presented in Figure 5, using K-reg, CTR2, and HYB-IBIM. Then we evaluate the function $\psi_{rxn}$ (2.31) in the origin to test the accuracy of the methods. Concurrently we test the area approximation with the three different Jacobian approximations used in the three different methods. The “diff” columns present the different of consecutive values in the column to their left. The “GMRES” column presents the number of GMRES iterations needed to solve the system.

| grid size | $A$, $J_r \approx 1$ | diff $\psi_{rxn}(0)$, K-reg | diff $\psi_{rxn}(0)$, CTR2 | diff $\psi_{rxn}(0)$, HYB-IBIM | GMRES |
|-----------|----------------------|-----------------------------|-----------------------------|---------------------------------|-------|
| 128$^3$   | 21.477653            | -2.39016e+02                |                             |                                 | 6     |
| 192$^3$   | 21.474494            | 3.159e-3                    | -2.39906e+02                | 8.89e-1                         | 7     |
| 256$^3$   | 21.473238            | 1.256e-3                    | -2.40361e+02                | 4.54e-1                         | 7     |
| 320$^3$   | 21.472567            | 6.709e-4                    | -2.40647e+02                | 2.86e-1                         | 7     |
| 512$^3$   | 21.472018            | 5.490e-4                    | -2.41084e+02                | 4.36e-1                         | 7     |

Table 5: Protein 1YCR. We solve the system on the SES of protein 1YCR using K-reg, CTR2, and HYB-IBIM, with width $\varepsilon = 2h$. Then we compute the polarization energy $\mathcal{G}_{pol}$ (3.3) to test the accuracy of the methods.

| grid size | $A$, $J_r \approx J_h$ | diff $\mathcal{G}_{pol}$, K-reg | diff $\mathcal{G}_{pol}$, CTR2 | diff $\mathcal{G}_{pol}$, HYB-IBIM | GMRES |
|-----------|------------------------|---------------------------------|---------------------------------|---------------------------------|-------|
| 128$^3$   | 4388.871               | -1.1905865e+03                 |                                 |                                 | 11    |
| 256$^3$   | 4461.273               | 7.24e+1                        | -1.1951553e+03                 | 4.56e+0                         | 12    |
| 512$^3$   | 4500.038               | 3.87e+1                        | -1.2025009e+03                 | 7.34e+0                         | 11    |
| 128$^3$   | 4366.619               | -1.1584408e+03                 |                                 |                                 | 15    |
| 256$^3$   | 4456.909               | 9.02e+1                        | -1.1824920e+03                 | 2.40e+1                         | 15    |
| 512$^3$   | 4499.230               | 4.23e+1                        | -1.1973823e+03                 | 1.48e+1                         | 15    |

| grid size | $A$, $J_r \approx J_h^{(H)}$ | diff $G_{pol}$, HYB-IBIM | GMRES |
|-----------|-------------------------------|--------------------------|-------|
| 128$^3$   | 4372.020                      | -1.1602900e+03           | 14    |
| 256$^3$   | 4457.470                      | 8.54e+1                  | 14    |
| 512$^3$   | 4499.400                      | 4.19e+1                  | 15    |

Acknowledgments

The authors would like to thank Zirui Zhang and Prof. Li-Tien Cheng for providing the smooth solvent-molecule interface for the p53-MDM2 protein via its signed distance function. The authors would also like to thank the Texas Advanced Computing Center (TACC) and the Swedish National Infrastructure for Computing at PDC Center for High Performance Computing for the computing.
resources. Tsai’s research is supported partially by NSF grant DMS-2110895.

References

[1] N. A. Baker. “Biomolecular applications of Poisson-Boltzmann methods”. In: Rev. Comput. Chem. 21 (2005), p. 349.

[2] B. Engquist, A.-K. Tornberg, and R. Tsai. “Discretization of Dirac delta functions in level set methods”. In: J. Comput. Phys. 207.1 (2005), pp. 28–51.

[3] M. Feig and C. L. Brooks III. “Recent advances in the development and application of implicit solvent models in biomolecule simulations”. In: Curr. Opin. Struct. Biol. 14.2 (2004), pp. 217–224.

[4] F. Fogolari, A. Brigo, and H. Molinari. “The Poisson–Boltzmann equation for biomolecular electrostatics: a tool for structural biology”. In: J. Mol. Recognit. 15.6 (2002), pp. 377–392.

[5] W. Fong and E. Darve. “The black-box fast multipole method”. In: J. Comput. Phys. 228.23 (2009), pp. 8712–8725.

[6] M. Gerstein, F. Richards, M. Chapman, and M. Connolly. “Protein surfaces and volumes: measurement and use”. In: ed. by M. Rossmann and E. Arnold. Vol. F. Crystallography of Biological Molecules. International Tables for Crystallography. Dordrecht, Netherlands: Kluwer, 2001. Chap. 22.1, pp. 531–545.

[7] Á. Helgadóttir and F. Gibou. “A Poisson–Boltzmann solver on irregular domains with Neumann or Robin boundary conditions on non-graded adaptive grid”. In: J. Comput. Phys. 230.10 (2011), pp. 3830–3848.

[8] F. Izzo, O. Runborg, and R. Tsai. “Convergence of a class of high order corrected trapezoidal rules”. In: arXiv preprint arXiv:2208.08216 (2022).

[9] F. Izzo, O. Runborg, and R. Tsai. “Corrected trapezoidal rules for singular implicit boundary integrals”. In: J. Comput. Phys. 461 (2022), p. 111193.

[10] F. Izzo, O. Runborg, and R. Tsai. “High order corrected trapezoidal rules for a class of singular integrals”. In: arXiv preprint arXiv:2203.04854 (2022).

[11] A. Juffer, E. F. Botta, B. A. van Keulen, A. van der Ploeg, and H. J. Berendsen. “The electric potential of a macromolecule in a solvent: A fundamental approach”. In: J. Comput. Phys. 97.1 (1991), pp. 144–171.

[12] C. Kublik, N. M. Tanushev, and R. Tsai. “An implicit interface boundary integral method for Poisson’s equation on arbitrary domains”. In: J. Comput. Phys. 247 (2013), pp. 279–311.

[13] C. Kublik and R. Tsai. “Integration over curves and surfaces defined by the closest point mapping”. In: Res. Math. Sci. 3.1 (2016), pp. 1–17.

[14] C. Kublik and R. Tsai. “An extrapolative approach to integration over hypersurfaces in the level set framework”. In: Math. Comp. (2018).

[15] P. H. Kussie, S. Gorina, V. Marechal, B. Elenbaas, J. Moreau, A. J. Levine, and N. P. Pavletich. “Structure of the MDM2 oncoprotein bound to the p53 tumor suppressor transactivation domain”. In: Science 274.5289 (1996), pp. 948–953.

[16] B. Li, X. Cheng, and Z. Zhang. “Dielectric boundary force in molecular solvation with the Poisson–Boltzmann free energy: A shape derivative approach”. In: SIAM J. Appl. Math. 71.6 (2011), pp. 2093–2111.

[17] B. Li and Y. Liu. “Diffused Solute-Solvent Interface with Poisson–Boltzmann Electrostatics: Free-Energy Variation and Sharp-Interface Limit”. In: SIAM J. Appl. Math. 75.5 (2015), pp. 2072–2092.

[18] B. Lu, Y. Zhou, M. Holst, and J. McAmmon. “Recent progress in numerical methods for the Poisson-Boltzmann equation in biophysical applications”. In: Commun. Comput. Phys. 3.5 (2008), pp. 973–1009.
[19] Z. Wang, J. Che, L.-T. Cheng, J. Dzubiella, B. Li, and J. A. McCammon. “Level-set variational implicit-solvent modeling of biomolecules with the Coulomb-field approximation”. In: J. Chem. Theory Comput. 8.2 (2012), pp. 386–397.

[20] Z. Zhang and L.-T. Cheng. “Binary Level Set Method for Variational Implicit Solvation Model”. In: arXiv preprint arXiv:2110.12815 (2021).

[21] Z. Zhang, C. G. Ricci, C. Fan, L.-T. Cheng, B. Li, and J. A. McCammon. “Coupling Monte Carlo, variational implicit solvation, and binary level-set for simulations of biomolecular binding”. In: J. Chem. Theory Comput. 17.4 (2021), pp. 2465–2478.

[22] Y. Zhong, K. Ren, and R. Tsai. “An implicit boundary integral method for computing electric potential of macromolecules in solvent”. In: J. Comput. Phys. 359 (2018), pp. 199–215.

[23] S. Zhou, L.-T. Cheng, J. Dzubiella, B. Li, and J. A. McCammon. “Variational implicit solvation with Poisson–Boltzmann theory”. In: J. Chem. Theory Comput. 10.4 (2014), pp. 1454–1467.

[24] S. Zhou, L.-T. Cheng, H. Sun, J. Che, J. Dzubiella, B. Li, and J. A. McCammon. “LS-VISM: A software package for analysis of biomolecular solvation”. In: J. Comput. Chem. 36.14 (2015), pp. 1047–1059.