Cleophas Kweyu*, Lihong Feng, Matthias Stein and Peter Benner

Reduced basis method for the nonlinear Poisson–Boltzmann equation regularized by the range-separated canonical tensor format

https://doi.org/10.1515/ijnsns-2021-0103

Received March 11, 2021; accepted April 26, 2022; published online May 17, 2022

Abstract: The Poisson–Boltzmann equation (PBE) is a fundamental implicit solvent continuum model for calculating the electrostatic potential of large ionic solvated biomolecules. However, its numerical solution encounters severe challenges arising from its strong singularity and nonlinearity. In (P. Benner, V. Khoromskaia, B. Khoromskij, C. Kweyu, and M. Stein, "Regularization of Poisson-Boltzmann type equations with singular source terms using the range-separated tensor format," *SIAM J. Sci. Comput.*, vol. 43, no. 1, pp. A415–A445, 2021; C. Kweyu, V. Khoromskaia, B. Khoromskij, M. Stein, and P. Benner, "Solution decomposition for the nonlinear Poisson-Boltzmann equation using the range-separated tensor format," arXiv:2109.14073, 2021), the effect of strong singularities was eliminated by applying the range-separated (RS) canonical tensor format (P. Benner, V. Khoromskaia, and B. N. Khoromskij, "Range-separated tensor format for many-particle modeling," *SIAM J. Sci. Comput.*, vol. 40, no. 2, pp. A1034–A1062, 2018; B. N. Khoromskij, "Range-separated tensor representation of the discretized multidimensional Dirac delta and elliptic operator inverse," *J. Comput. Phys.*, vol. 401, p. 108998, 2020) to construct a solution decomposition scheme for the PBE. The RS tensor format allows deriving a smooth approximation to the Dirac delta distribution in order to obtain a regularized PBE (RPBE) model. However, solving the RPBE is still computationally demanding due to its high dimension $\mathcal{N}$, where $\mathcal{N}$ is always in the millions. In this study, we propose to apply the reduced basis method (RBM) and the (discrete) empirical interpolation method ((D)EIM) to the RPBE in order to construct a reduced order model (ROM) of low dimension $N \ll \mathcal{N}$, whose solution accurately approximates the nonlinear RPBE. The long-range potential can be obtained by lifting the ROM solution back to the $\mathcal{N}$-space while the short-range potential is directly precomputed analytically, thanks to the RS tensor format. The sum of both provides the total electrostatic potential. The main computational benefit is the avoidance of computing the numerical approximation of the singular electrostatic potential. We demonstrate in the numerical experiments, the accuracy and efficacy of the reduced basis (RB) approximation to the nonlinear RPBE (NRPBE) solution and the corresponding computational savings over the classical nonlinear PBE (NPBE) as well as over the RBM being applied to the classical NPBE.

Keywords: canonical tensor representation; Dirac delta distribution; Newton kernel; range-separated tensor format; reduced basis method; singularity.

AMS Subject Classification: 65F30; 65F50; 65N35; 65F10.

***Corresponding author: Cleophas Kweyu**, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, D-39106 Magdeburg, Germany; and Department of Mathematics and Physics, Moi University, P.O. Box 3900-30100, Eldoret, Kenya, E-mail: kweyuc@mu.ac.ke

Lihong Feng and Matthias Stein, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, D-39106 Magdeburg, Germany, E-mail: feng@mpi-magdeburg.mpg.de (L. Feng), matthias.stein@mpi-magdeburg.mpg.de (M. Stein)

Peter Benner, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, D-39106 Magdeburg, Germany; and Faculty of Mathematics, Otto von Guericke University, Magdeburg, Germany, E-mail: benner@mpi-magdeburg.mpg.de

1 Introduction

The Poisson–Boltzmann equation (PBE) is a second order nonlinear elliptic partial differential equation (PDE) which is ubiquitous in the modeling of biochemical processes [1–4]. It is used to calculate the electrostatic potential throughout the biomolecular system consisting of the biomolecule and the surrounding ionic or salt solution. More information about the significance of the electrostatic interactions and the related PBE post-processing, for instance, the electrostatic forces and energies, can be found in [5–7]. Figure 1 illustrates the two dimensional (2D) view of the biomolecular system consisting of a low dielectric molecular region (Ω_m) encapsulated by an ionic solution of high dielectric[1] in Ω_s.

One of the fundamental applications of electrostatic potential computations is the Brownian dynamics simulation which may be used for example, to determine protein association rates and to simulate protein–protein encounter [5]. Protein association rates highly depend on the ionic strength of the solution. For instance, high ionic strengths dampen or attenuate the effect of electrostatic forces and energies of proteins, hence reducing the rates of association and vice versa. Therefore, in this work we consider the solution of the PBE parameterized by the ionic strength.

The analytical solution to the PBE for biomolecules with complex geometries, strong nonlinearities, and highly singular charge density distributions is not available [8, 9]. To this end, numerical methods, for example, the finite difference method (FDM) [10, 11], the finite element method (FEM) [10, 12], the boundary element method (BEM) [13, 14], are widely used to solve the PBE. Interested readers are referred to [4] for a thorough review of the aforementioned techniques for solving the PBE. However, the numerical solution to the PBE faces various challenges. The most severe are: the strong singularities, caused by the Dirac delta distribution sources; the strong nonlinearity, caused by the exponential nonlinear terms; the unbounded domain, due to the slow polynomial decay in the form of $1/\|\bar{x}\|$ as $\|\bar{x}\| \to \infty$; and imposing the correct jump or interface conditions to the rather irregular molecular domain, Γ [15–17].

In [18, 19], the strong singularities are circumvented by using the range-separated (RS) canonical tensor format, which was introduced and analyzed in [20]. Consequently, a nonlinear regularized PBE (NRPBE) model is determined, which only solves for smooth long-range electrostatic potential. The jump conditions are annihilated due to the accurate splitting of the long- and short-range components of the total electrostatic potential using the RS tensor format. Nevertheless, the computational cost of solving the regularized nonlinear PBE is still high due to its high dimension $\mathcal{N} \approx \mathcal{O}(10^6)$. In this work, we apply the reduced basis method (RBM), in order to construct an accurate reduced order model (ROM) of much lower dimension, i.e., of $\mathcal{O}(10)$ for the NRPBE. The simulations for varying parameter values, in this case, the varying ionic strength, can be

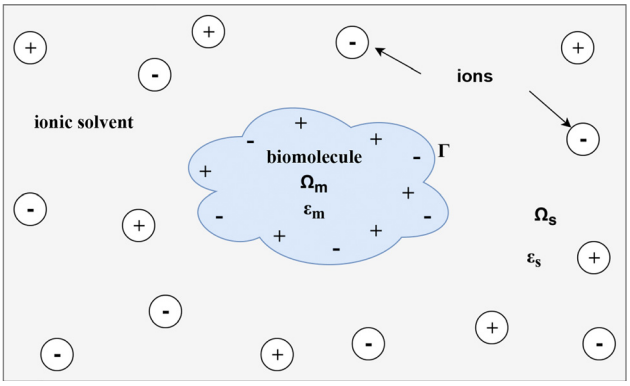


Figure 1: Schematic representation of the electrostatic interactions of a biomolecule (m) with an atomic charge distribution and dielectric constant ϵ_m, in a solution (s) of ions with a dielectric constant ϵ_s.

1 Here, dielectric (or electric permittivity) refers to the ability of a material to transmit electric force without conduction, [52].

computed much faster by using the parametrized ROM and hence the Brownian dynamics simulations can be significantly accelerated.

The RBM can also be directly applied to the classical NPBE without any regularization. We demonstrate that applying the RBM to the NRPBE gives rise to more robust and efficient solution to the problem as compared to applying the RBM to the classical NPBE.

It is worth noting that the RBM has been applied to a simplified variant of the classical nonlinear PBE in [21] in 1 and 2 dimensions wherein smooth exponential functions were used as the source terms. In this work, we apply the RBM to the 3-dimensional NRPBE for biomolecular simulation of large (complex) solvated biomolecules, for example, a protein in an ionic solutions, whose electrostatic potential $u(\bar{x})$ is characterized by the slow polynomial decay in $1/\|\bar{x}\|$, i.e., $u(\infty) = 0$, hence large domains have to be considered for accurate approximation of boundary conditions, resulting in high dimension of $\mathcal{O}(10^6)$ for the discretized system [19]. Furthermore, we consider Dirichlet boundary conditions which are nonaffine in parameter, thereby requiring the application of (D)EIM in order to reduce the complexity of the boundary conditions, and in turn, to further reduce the ROM complexity [22–24].

The main contributions of this paper include: we have applied a more efficient numerical method for solving the NRPBE, which is based on first linearization via the Taylor series truncation of the nonlinear term, followed by discretization. This approach avoids the computation of the Jacobian of a huge matrix and also converges much faster than the standard Newton iteration; we have successfully applied the RBM and (D)EIM in order to reduce the dimension and complexity of the NRPBE; numerical comparison of RBM applied to the NRPBE and the classical NPBE shows that the former is more efficient and accurate.

We highlight the main contributions of this paper as follows:

- We construct the state of the art range-separated (RS) tensor-based nonlinear PBE model by the RS tensor format that was developed in [20, 25]. This results in the efficient splitting of the electrostatic potential into two parts, the short-range and long-range parts, within the molecular and solvent regions, respectively. This avoids the use of some cutoff function of the Laplace operator at the solute-solvent interface that is inherent in the existing solution decomposition techniques, in which the jump/interface conditions depend on the short-range (singular) electrostatic potential.
- We solve the high fidelity full order model using an efficient numerical technique, which is based on a linearization using the Taylor series truncation of the nonlinear term followed by discretization. This approach avoids the computation of the Jacobian of a huge matrix and also converges much faster than the standard Newton iteration.
- We for the very first time apply the reduced basis method (RBM), a parametrized model reduction tool and the (discrete) empirical interpolation method ((D)EIM) to both the classical and the regularized nonlinear parametric PBE models in order to provide greatly reduced computational costs (i.e., time and storage) in multi-query contexts, where the ionic strength is varied. We notice that due to the absence of solution singularities in the regularized PBE model, the reduced order model (ROM) is of much lower dimension than that for the classical model and of higher accuracy.

The remainder of the paper is structured as follows. Section 3 briefly reviews the approach of regularizing the PBE model by the RS canonical tensor format as proposed in [18, 19]. In Section 4.2, the RBM framework and its application to both the regularized PBE (RPBE) and the classical PBE is introduced. Finally, Section 5 presents the numerical experiments to illustrate the computational advantages of the RBM for the RPBE over the classical PBE. Comparisons with the solutions obtained by the standard FDM-based PBE solvers for the classical PBE are also presented.

2 Mathematical model of the PBE

The nonlinear PBE for a symmetric 1:1 salt is given by

$$-\nabla \cdot (\epsilon(\bar{x})\nabla u(\bar{x})) + \bar{\kappa}^2(\bar{x})\sinh(u(\bar{x})) = \sum_{i=1}^{N_m} q_i \delta(\bar{x} - \bar{x}_i), \quad \Omega \in \mathbb{R}^3, \tag{2.1}$$

subject to

$$u(\bar{x}) = \frac{1}{4\pi}\sum_{i=1}^{N_m} \frac{q_i e^{-\kappa(d-a_i)}}{\epsilon_s(1+\kappa a_i)d} \quad \text{on the boundary } \partial\Omega, \quad d = \|\bar{x} - \bar{x}_i\|, \quad \bar{x} = (x, y, z), \tag{2.2}$$

where $q_i = 4\pi e_c^2 z_i/\kappa_B T$, z_i is the partial charge of each atom, $u(\bar{x})$ represents the dimensionless potential, $\kappa_B T$, κ_B, T, and e_c are the thermal energy, the Boltzmann constant, the absolute temperature, and the electron charge, respectively. The Debye–Hückel screening parameter, $\kappa^2 = 8\pi N_A e_c^2 I/1000\epsilon_s\kappa_B T$, describes ion concentration and accessibility, ϵ_s is the solvent dielectric coefficient, a_i is the atomic radius, and N_m is the sum of the partial charges in the biomolecule. The sum of Dirac delta distributions represents the highly singular molecular charge density.

The dielectric coefficient $\epsilon(\bar{x})$ and kappa function $\bar{\kappa}^2(\bar{x})$ are piecewise constant functions given by

$$\epsilon(\bar{x}) = \begin{cases} \epsilon_m & \text{if } \bar{x} \in \Omega_m, \\ \epsilon_s & \text{if } \bar{x} \in \Omega_s, \end{cases} \quad \bar{\kappa}(\bar{x}) = \begin{cases} 0 & \text{if } \bar{x} \in \Omega_m, \\ \sqrt{\epsilon_s}\kappa & \text{if } \bar{x} \in \Omega_s, \end{cases} \tag{2.3}$$

where Ω_m and Ω_s are the regions occupied by the protein molecule and by the ionic solution, respectively, as shown in Figure 1. E.g., for the examples reported in this paper, we set $\epsilon_m = 2$ and $\epsilon_s = 78.54$. See [8, 26, 27] for discussions regarding the PBE theory and the importance of (2.1) in biomolecular modeling. Please note that (2.1) has to be understood in the weak derivative sense due to the jump of the coefficients ϵ and $\overline{\kappa}$ across the interface. We will detail our approach to deal with this discontinuity in Section 3.

The PBE in (2.1) can be linearized for small electrostatic potentials by retaining the first term of the Taylor series expansion of the nonlinear function $\sinh(u(\bar{x}))$ [28]. The LPBE is thus given by

$$-\nabla \cdot (\epsilon(\bar{x})\nabla u(\bar{x})) + \bar{\kappa}^2(\bar{x})u(\bar{x}) = \sum_{i=1}^{N_m} q_i \delta(\bar{x} - \bar{x}_i). \tag{2.4}$$

The LPBE is much easier to solve and very accurate for lowly charged biomolecules, for example, proteins. However, for highly charged biomolecules, such as nucleic acids, it is not as accurate as the nonlinear variant due to the magnitude of the electric field at the interface between the solute and the solvent, Γ [5, 29].

3 Regularization of the PBE by the RS tensor format

The numerical approximation of the PBE is hindered by the highly singular sources described by a sum of Dirac delta distributions. This is because, for every singular charge z_i in (2.1), there corresponds degenerate behaviour in the electrostatic potential $u(\bar{x}_i)$ at each atomic position $\bar{x}_i$ in Ω_m. To circumvent this drawback, various researchers have developed solution decomposition approaches for the PBE [15–17, 30]. A common feature of these approaches is that they circumvent the building of numerical approximations corresponding to the Dirac delta distributions by solving a regularized PBE model for the smooth long-range electrostatic potential. This is enhanced by the fact that analytical expansions by the Newton kernel are possible in the solute sub-region Ω_m.

In principle, the solution decomposition techniques for the PBE involve coupling of two equations for the electrostatic potential in the solute and solvent regions, through the interface, Γ [16, 30]. Due to the absence of ions within the molecular region Ω_{m}, it is modeled by the Poisson equation,

$$-\nabla \cdot (\epsilon_{\mathrm{m}} \nabla u) = \sum_{i=1}^{N_{\mathrm{m}}} q_i \delta(\bar{x} - \bar{x}_i) \quad \text{in } \Omega_{\mathrm{m}}. \tag{3.1}$$

On the other hand, no atoms are present in the solvent region Ω_{s}, hence the charge density is purely modeled by the Boltzmann distribution, leading to

$$-\nabla \cdot (\epsilon_{\mathrm{s}} \nabla u) + \bar{\kappa}^2 \sinh(u) = 0 \quad \text{in } \Omega_{\mathrm{s}}. \tag{3.2}$$

Therefore, the two Eqs. (3.1) and (3.2) are coupled together via the jump (interface) boundary conditions

$$[u]_\Gamma = 0, \quad \text{and} \quad \left[\epsilon \frac{\partial u}{\partial n_\Gamma}\right]_\Gamma = 0, \tag{3.3}$$

where $\Gamma := \partial\Omega_{\mathrm{m}} = \partial\Omega_{\mathrm{s}} \cap \Omega_{\mathrm{m}}$ and $[f]_\Gamma = \lim_{t\to 0} \left(f(\bar{x} + tn_\Gamma) - f(\bar{x} - tn_\Gamma)\right)$. Here, we denote n_Γ as the unit outward normal direction of the interface Γ.

In [18, 19], the authors employ the RS canonical tensor format, developed and analyzed in [20], to construct the solution decomposition of the PBE. This is realized by approximating the singular sources with a smooth function derived from the long-range component of the Newton potential sum. The resultant regularized PBE solves for the long-range electrostatic potential, which is then added to the short-range component that is precomputed from the RS tensor splitting of the Newton kernel. The regularized PBE (RPBE) model has demonstrated to be much more accurate than the classical PBE model in [18, 19]. We highlight the core ingredients for obtaining the RPBE in Section 3.1.

3.1 Canonical tensor representation of the Newton kernel

Definition 3.1. The Newton potential of an integrable function (or a Radon measure) f with compact support in $\mathbb{R}^3$ is defined as the convolution

$$u(\bar{x}) = \Gamma_N * f(\bar{x}) = \int_{\mathbb{R}^3} \Gamma_N(\bar{x} - \bar{y}) f(\bar{y}) \mathrm{d}y, \tag{3.4}$$

where the Newton kernel $\Gamma_N = 1/\|\bar{x}\|$, has a mathematical singularity at the origin, and $\bar{y} \in \mathbb{R}^3$ [31]. The Newton potential $u(\bar{x})$ satisfies the Poisson equation

$$-\Delta u = f, \tag{3.5}$$

where f in this case is the source term of the system as defined in (3.7).

Consider the single particle Newton potential (or the Newton kernel) $1/\|\bar{x}\|$, $\bar{x} \in \mathbb{R}^3$, which is a fundamental solution to the Poisson equation. It is well known that determining a weighted sum of interaction potentials (or Newton kernels), $P_N(\bar{x})$ in a large N_{m}-particle system, with the particle locations at $\bar{x}_i \in \mathbb{R}^3$, $i = 1, \ldots, N_{\mathrm{m}}$, i.e.,

$$P_N(\bar{x}) = \sum_{i=1}^{N_m} \frac{q_i}{\epsilon_m \|\bar{x} - \bar{x}_i\|}, \quad \bar{x}_i,\ \bar{x} \in \Omega = [-b, b]^3, \tag{3.6}$$

is quite computationally demanding. The Newton kernel exhibits a slow polynomial decay in $1/\|\bar{x}\|$ as $\|\bar{x}\| \to \infty$. Obviously, it has a singularity at $\bar{x} = (0, 0, 0)$, making its accurate grid representation difficult. The

RS tensor format [20] can be exploited to construct an efficient grid-based technique for the calculation of $P_N(\bar{x})$ in multiparticle systems.

Remark 1. Notice that the Newton potential $P_N(\bar{x})$ in (3.6) is a special case of (3.4) for a noncompact function

$$f(x) = \frac{1}{\epsilon_m} \sum_{i=1}^{N_m} q_i \delta(\bar{x} - \bar{x}_i). \tag{3.7}$$

To obtain the canonical tensor representation of the Newton kernel, we follow the procedure in [20], whereby we first consider the computational domain $\Omega = [-b, b]^3$, and introduce the uniform $(n^{\otimes 3})^2$ rectangular Cartesian grid Ω_n with mesh size $h = 2b/n$ (n even). Let $\{\psi_i\}$ be a set of tensor-product piecewise constant basis functions, $\psi_i(\bar{x}) = \prod_{\ell=1}^{3} \psi_{i_\ell}^{(\ell)}(\bar{x}_\ell)$, for the 3-tuple index $i = (i_1, i_2, i_3)$, $i_\ell \in I_\ell = \{1, \dots, n\}$, $\ell = 1, 2, 3$. The goal is to discretize the Newton kernel by its projection onto $\{\psi_i\}$ as follows

$$\mathbf{P} := [p_i] \equiv [p(i_1, i_2, i_3)] \in \mathbb{R}^{n^{\otimes 3}}, \quad p_i = \int_{\mathbb{R}^3} \frac{\psi_i(\bar{x})}{\|\bar{x}\|} \, \mathrm{d}\bar{x}, \tag{3.8}$$

where p_i is obtained from the vectors of the canonical tensor representation of the Newton kernel.

Next, determine the Laplace–Gauss transform representation of $1/\|\bar{x}\|$, and then apply the exponentially convergent sinc-quadrature approximation to obtain the separable expansion

$$\frac{1}{\|\bar{x}\|} = \frac{2}{\sqrt{\pi}} \int_{\mathbb{R}^+} \mathrm{e}^{-t^2\|\bar{x}\|^2} \mathrm{d}t \approx \sum_{k=-M}^{M} a_k \mathrm{e}^{-t_k^2\|\bar{x}\|^2} = \sum_{k=-M}^{M} a_k \prod_{\ell=1}^{3} \mathrm{e}^{-t_k^2 \bar{x}_\ell^2}, \tag{3.9}$$

where the quadrature points and weights in (3.9) are given by

$$t_k = k\mathfrak{h}_M, \quad a_k = 2\mathfrak{h}_M/\sqrt{\pi}, \quad \text{with } \mathfrak{h}_M = C_0 \log(M)/M, \quad C_0 \approx 3. \tag{3.10}$$

The mode three tensor $\mathbf{P}$, can be approximated by the R-term ($R = 2M + 1$) canonical tensor representation

$$\mathbf{P} \approx \mathbf{P}_{\mathrm{R}} = \sum_{k=-M}^{M} \mathbf{p}_k^{(1)} \otimes \mathbf{p}_k^{(2)} \otimes \mathbf{p}_k^{(3)} \in \mathbb{R}^{n^{\otimes 3}}, \tag{3.11}$$

where $\mathbf{p}_k^{(\ell)} \in \mathbb{R}^n$ are obtained by substituting (3.9) into (3.8) and "$\otimes$" [3] denotes the outer (or tensor) product of vectors.

Notice that the sum in (3.11) reduces to $k = 0, 1, \dots, M$ due to the symmetry argument of the Newton kernel. The sequence of quadrature points $\{t_k\}$ in (3.10) can be split into two subsequences $\mathcal{T} = \{t_k | k = 0, 1, \dots, M\} = \mathcal{T}_1 \cup \mathcal{T}_s$, where $\mathcal{T}_1 := \{t_k | k = 0, 1, \dots, R_1\}$ includes quadrature points t_k condensed near zero, thereby generating the long-range Gaussians and $\mathcal{T}_s := \{t_k | k = R_l + 1, \dots, M\}$ accumulates the increasing in $M \to \infty$ sequence of large sampling points t_k corresponding to the short-range Gaussians, with an upper bound of $C_0^2 \log^2(M)$. For more details, see [19, 20].

Upon splitting the reference canonical tensor representation $\mathbf{P}_R$ by the procedure presented in [20], we obtain the following decomposition

$$\mathbf{P}_{\mathrm{R}} = \mathbf{P}_{\mathrm{R}_s} + \mathbf{P}_{\mathrm{R}_1},$$

where

$$\mathbf{P}_{\mathrm{R}_s} = \sum_{k \in \mathcal{K}_s} \mathbf{p}_k^{(1)} \otimes \mathbf{p}_k^{(2)} \otimes \mathbf{p}_k^{(3)}, \quad \mathbf{P}_{\mathrm{R}_1} = \sum_{k \in \mathcal{K}_1} \mathbf{p}_k^{(1)} \otimes \mathbf{p}_k^{(2)} \otimes \mathbf{p}_k^{(3)}. \tag{3.12}$$

Here, $\mathcal{K}_1 := \{k | k = 0, 1, \dots, R_1\}$ and $\mathcal{K}_s := \{k | k = R_1 + 1, \dots, M\}$ are the sets of indices for the long- and short-range canonical vectors.

2 $n^{\otimes 3} = n \times n \times n$ is a tensor representation of the 3D Cartesian grid.

The critical number R_l which specifies the splitting $\mathcal{T} = \mathcal{T}_l \cup \mathcal{T}_s$ is determined by the active support of the short-range components. In this case, one can cut off the functions $\mathbf{p}_k(x)$, $t_k \in \mathcal{T}_s$, outside of the sphere B_σ of radius $\sigma > 0$ subject to a certain threshold accuracy $\delta > 0$. The cross-sectional view of the respective localized and global vector components of the Newton potential in (3.12) on the x-axis is illustrated in Figure 2.

The aforementioned results are only valid for a single particle potential (or the Newton kernel, $1/\|\bar{x}\|$). In the case of a potential sum generated by a multiparticle system, for example, in (3.6), the two components in (3.12) are treated independently due to their differences as far as their effective supports are concerned [20]. The following is an overview of the RS canonical tensor representation of $P_N(\bar{x})$ in (3.6).

We first consider the tensor representation of only the long-range component $\mathbf{P}_{R_l} \in \mathbb{R}^{n^{\otimes 3}}$ which can be constructed by a direct sum of shift-and-windowing transforms, $\mathcal{W}_i$, of the reference tensor $\widetilde{\mathbf{P}}_{R_l} \in \mathbb{R}^{2n\times 2n\times 2n}$ from a large $(2n)^{\otimes 3}$ domain onto the original $n^{\otimes 3}$ domain. See [20, 32] for detailed information.

$$\mathbf{P}_l = \sum_{i=1}^{N_m} z_i \, \mathcal{W}_i(\widetilde{\mathbf{P}}_{R_l}) = \sum_{i=1}^{N_m} z_i \, \mathcal{W}_i \left(\sum_{k\in\mathcal{K}_l} \widetilde{\mathbf{p}}_k^{(1)} \otimes \widetilde{\mathbf{p}}_k^{(2)} \otimes \widetilde{\mathbf{p}}_k^{(3)} \right). \tag{3.13}$$

Remark 2. Note that $\mathbf{P}_l$ comprises of a collection of $\mathbf{P}_{R_l}$ at each atomic position in the entire protein, which have been shifted and windowed by the transform $\mathcal{W}_i$ of the reference tensor $\widetilde{\mathbf{P}}_{R_l} \in \mathbb{R}^{2n\times 2n\times 2n}$. Clearly, $\widetilde{\mathbf{P}}_{R_l}$ consists of $\mathbf{P}_{R_l}$ in a $(2n)^{\otimes 3}$ domain.

The reference tensor $\widetilde{\mathbf{P}}_{R_l}$ is mapped onto its subtensor of smaller size $n^{\otimes 3}$, by first shifting the center of $\widetilde{\mathbf{P}}_{R_l}$ to the grid-point x_i, and then windowing (restricting) the result onto the computational grid Ω_n. The particle charges are denoted by z_i. The canonical rank of the tensor sum $\mathbf{P}_l$, of rank RN_m, was proven in [20] to depend only logarithmically on the number of particles N_m involved in the summation.

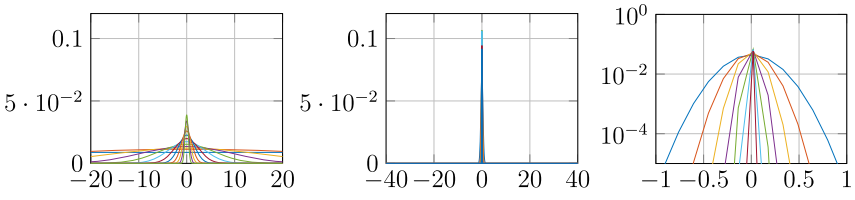
Remark 3. It is worth noting that for large biomolecules, the rank RN_m and the $n^{\otimes 3}$ Cartesian grid can be very large due to large N_m. In such cases, the canonical-to-Tucker (C2T) and the Tucker-to-canonical (T2C) transforms can be applied in order to obtain a low rank canonical tensor representation which accurately approximates the original tensor. The C2T transform employs the reduced higher order singular value decomposition (RHOSVD) to accomplish the rank reduction process [33].

On the other hand, the short-range part of the total electrostatic potential is represented by a single small size tensor $\mathbf{P}_s \in \mathbb{R}^{n^{\otimes 3}}$, known as the cumulated canonical tensors (CCT) [20]. The CCT comprises of localized subtensors whose effective supports are nonintersecting

$$\mathbf{P}_s = \sum_{i=1}^{N_m} z_i \mathbf{U}_i, \qquad \mathbf{U}_i \in \mathbb{R}^{n_s^{\otimes 3}}, \quad n_s \ll n, \tag{3.14}$$

where $\operatorname{diam}(\operatorname{supp}\mathbf{U}_i) \le 2\sigma_i$. Here, σ_i is the atomic radius of each atom in the biomolecule.

Remark 4. Notice that for biomolecules whose atoms have varying radii, we adjust the computation of the short- and long-range range electrostatic potential accordingly by assigning the corresponding vectors from Figure 2 to atomic clusters of similar radii [19].



(a) Long-range vectors. **(b)** Short-range vectors. **(c)** Log plot for Figure 2b.

Figure 2: Canonical vector components for the long-range and short-range electrostatic potentials and the logarithmic plot for magnifying the short-range vectors.

3.2 Construction of the nonlinear RPBE (NRPBE)

We now have sufficient information to facilitate the construction of the NRPBE based on the simple splitting of the Dirac delta distribution [25]. To fix the idea, from Remark 1, the weighted sum of interaction potentials in a large N_{m}-particle system as in (3.6) is also the analytical solution to the Poisson equation (PE), i.e.,

$$-\epsilon_{\mathrm{m}}\Delta P_N(\bar{x}) = \sum_{i=1}^{N_{\mathrm{m}}} q_i\delta(\bar{x}-\bar{x}_i) \quad \text{in } \mathbb{R}^3. \tag{3.15}$$

Consider the RS tensor splitting of the multiparticle Newton potential into a sum of long-range tensors $\mathbf{P}_{\mathrm{l}}$ in (3.13) and a CCT tensor $\mathbf{P}_{\mathrm{s}}$ in (3.14), i.e.,

$$\mathbf{P}_{\mathrm{N}}(\bar{x}) = \mathbf{P}_{\mathrm{s}}(\bar{x}) + \mathbf{P}_{\mathrm{l}}(\bar{x}). \tag{3.16}$$

Applying the discretized Laplacian operator to each component of $P_{\mathrm{N}}(\bar{x})$, we obtain,

$$f^{\mathrm{s}} := -A_{\Delta}\mathbf{P}_{\mathrm{s}}, \quad \text{and} \quad f^{\mathrm{l}} := -A_{\Delta}\mathbf{P}_{\mathrm{l}}, \tag{3.17}$$

where A_{Δ} is the 3D finite difference Laplacian matrix defined on the uniform rectangular grid as

$$A_{\Delta} = \Delta_1 \otimes I_2 \otimes I_3 + I_1 \otimes \Delta_2 \otimes I_3 + I_1 \otimes I_2 \otimes \Delta_3, \tag{3.18}$$

where $-\Delta_{\ell} = \epsilon_m h_{\ell}^{-2}\mathrm{tridiag}\{1,-2,1\} \in \mathbb{R}^{n_{\ell}\times n_{\ell}}$, $\ell = 1,2,3$, denotes the discrete univariate Laplacian and I_{ℓ}, $\ell = 1,2,3$, is the identity matrix in each dimension. See [18, 19, 25] for more details. Thus, we have

$$f^{\mathrm{s}} + f^{\mathrm{l}} = \sum_{i=1}^{N_m} q_i\delta(\bar{x}-\bar{x}_i). \tag{3.19}$$

The nonlinear regularized PBE (NRPBE) can now be derived as an approximation of the PBE (2.1) as follows

$$\begin{cases} -\nabla\cdot(\epsilon\nabla(u)) + \bar{\kappa}^2\sinh(u) = f^{\mathrm{s}} + f^{\mathrm{l}} \text{ in } \Omega, \\ \qquad\qquad\qquad\qquad u = g \qquad \text{on } \partial\Omega, \end{cases} \tag{3.20}$$

along with the interface conditions of (3.3). Here, g denotes the boundary value function given in (2.2).

Using the splitting (3.19) of the Dirac-delta distributions, we split the solution u of (3.20) as

$$u = u^{\mathrm{s}} + u^{\mathrm{l}},$$

where u^{s} denotes the short-range component that satisfies the Poisson equation,

$$\begin{cases} -\nabla\cdot(\epsilon\nabla u^{\mathrm{s}}) = f^{\mathrm{s}} \quad \text{in } \Omega, \\ \qquad\qquad u^{\mathrm{s}} = 0 \quad \text{on } \partial\Omega, \end{cases} \tag{3.21}$$

together with the interface conditions on the interface Γ,

$$[u^{\mathrm{s}}]_{\Gamma} = 0, \quad \text{and} \quad \left[\epsilon\frac{\partial u^{\mathrm{s}}}{\partial n_{\Gamma}}\right]_{\Gamma} = 0, \tag{3.22}$$

and u^{l} denotes the long-range component function defined by the nonlinear boundary value problem

$$\begin{cases} -\nabla\cdot(\epsilon\nabla u^{\mathrm{l}}(\bar{x})) + \bar{\kappa}^2(\bar{x})\sinh(u^{\mathrm{l}}(\bar{x})) = f^{\mathrm{l}} \quad \text{in } \Omega, \\ \qquad\qquad\qquad\qquad u^{\mathrm{l}} = g \quad \text{on } \partial\Omega, \end{cases} \tag{3.23}$$

subject to the interface conditions in (3.22).

Here we have used that

$$\bar{\kappa}^2\sinh(u^{\mathrm{s}} + u^{\mathrm{l}}) = \bar{\kappa}^2\sinh(u^{\mathrm{l}}), \tag{3.24}$$

because it was proved and demonstrated in [18] that the function f^{s} and the corresponding short-range potential u^{s} are localized within the molecular region Ω_{m} and vanish on the interface Γ.

4 Numerical approach to solving the NRPBE

4.1 Iterative solution of the NRPBE

Let us consider a physical domain $\Omega \subset \mathbb{R}^3$ with boundary $\partial\Omega$, and a parameter domain $\mathcal{P} \subset \mathbb{R}$ which represents the variation in ionic strength $I = 1/2\sum_{j=1}^{N_{\mathrm{ions}}} c_j z_j^2$, which is a function of the ionic concentration c_i, of the salt solution. It resides in $\bar{k}^2 = 8\pi e^2 I/1000\epsilon k_{\mathrm{B}}T$. One standard way of solving the NRPBE in (3.23) is that it is first discretized in space to obtain a nonlinear system in vector form

$$a(u^{\mathrm{r}}_{\mathcal{N}}(\mu)) = b^{\mathrm{r}}(\mu), \quad \mu \in \mathcal{P}, \tag{4.1}$$

where $a(u^{\mathrm{r}}_{\mathcal{N}}(\mu)) \in \mathbb{R}^{\mathcal{N}}$, $b^{\mathrm{r}}(\mu) \in \mathbb{R}^{\mathcal{N}}$, $\mu = I \in \mathcal{P}$, and $u^{\mathrm{r}}_{\mathcal{N}}(\mu)$ is the discretized solution vector.

Then system (4.1) can be solved using several existing techniques. For example, nonlinear relaxation methods have been implemented in the Delphi software [34], the nonlinear conjugate gradient (CG) method has been implemented in University of Houston Brownian Dynamics (UHBD) software [35], the nonlinear multigrid (MG) method [36] and the inexact Newton method are available in the adaptive Poisson–Boltzmann solver (APBS) software [37].

In this study, we apply a different approach of solving (3.23) [17, 21, 38]. In particular, an iterative approach is first applied to the continuous NRPBE in (3.23), where at the $(n+1)$st iteration step, the NRPBE is approximated by a linear equation via the Taylor series truncation. The expansion point of the Taylor series is the continuous solution $(u^{\mathrm{l}}(\mu))^n$ at the nth iteration step.

Consider $(u^{\mathrm{l}}(\mu))^n$ as the approximate solution at the nth iterative step, then the nonlinear term $\sinh((u^{\mathrm{l}}(\mu))^{n+1})$ at the $(n+1)$st step is approximated by its truncated Taylor series expansion as follows

$$\sinh((u^{\mathrm{l}}(\mu))^{n+1}) \approx \sinh((u^{\mathrm{l}}(\mu))^n) + ((u^{\mathrm{l}}(\mu))^{n+1} - (u^{\mathrm{l}}(\mu))^n)\cosh((u^{\mathrm{l}}(\mu))^n). \tag{4.2}$$

Substituting the approximation (4.2) into (3.23), we obtain

$$\begin{aligned} &-\nabla\cdot(\epsilon(\bar{x})\nabla(u^{\mathrm{l}}(\mu))^{n+1}) + \bar{\kappa}^2(\bar{x})\cosh((u^{\mathrm{l}}(\mu))^n)(u^{\mathrm{l}}(\mu))^{n+1} \\ &\quad = -\bar{\kappa}^2(\bar{x})\sinh((u^{\mathrm{l}}(\mu))^n) + \bar{\kappa}^2(\bar{x})\cosh((u^{\mathrm{l}}(\mu))^n)(u^{\mathrm{l}}(\mu))^n + b^{\mathrm{r}}(\mu). \end{aligned} \tag{4.3}$$

The equation in (4.3) is linear, and can then be numerically solved by first applying spatial discretization. In this regard, we first define

$$\cosh\odot u^{\mathrm{r}}_{\mathcal{N}}(\mu) =: w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{\mathcal{N}} \end{bmatrix}, \tag{4.4}$$

where $\odot$ is the elementwise operation on a vector.

Then, we construct the corresponding diagonal matrix from (4.4) of the form

$$B = \mathrm{diag}(w_1, w_2, \ldots, w_{\mathcal{N}}).$$

Finally, we obtain the following iterative linear system

$$A_1(u^{\mathrm{r}}_{\mathcal{N}}(\mu))^{n+1} + \mu A_2 B^n(u^{\mathrm{r}}_{\mathcal{N}}(\mu))^{n+1} = -\mu A_2\ \sinh\odot(u^{\mathrm{r}}_{\mathcal{N}}(\mu))^n + \mu A_2 B^n(u^{\mathrm{r}}_{\mathcal{N}}(\mu))^n + b^{\mathrm{r}}_1 + b_2(\mu), \tag{4.5}$$

where A_1 is the Laplacian matrix and A_2 is a diagonal matrix containing the net $\bar{\kappa}^2$ function (i.e., $\bar{\kappa}^2/\mu$). Note that the diagonal matrix B^n changes at each iteration step, therefore, it cannot be precomputed. The vectors b^{r}_1

and $b_2(\mu)$ are the regularized approximation of the Dirac delta distributions and the nonaffine (in parameter) Dirichlet boundary conditions, respectively.

Let

$$A(\cdot) = A_1 + \mu A_2 B^n(\cdot) \tag{4.6}$$

and

$$F : \text{right} - \text{hand side of (4.5)}, \tag{4.7}$$

we obtain

$$A(u^{\mathrm{r}}_{\mathcal{N}}(\mu))^{n+1} = F((u^{\mathrm{r}}_{\mathcal{N}}(\mu))^n), \quad n = 0, 1, \ldots. \tag{4.8}$$

Then, at each iteration, system (4.8) is a linear system w.r.t. $\left(u^{\mathrm{l}}_{\mathcal{N}}\right)^{n+1}$, which can be solved by any linear system solver of choice. In this study, we employ the aggregation-based algebraic multigrid method (AGMG)[3] [39]. Algorithm 1 summarizes the detailed iterative approach for solving (4.8). This approach of first linearization, then discretization is shown to be more efficient than the standard way of first discretization and then linearization, via, for example, the Newton iteration. The advantage of the proposed approach is that it avoids computing the Jacobian of a huge matrix. It is observed that it converges faster than the standard Newton approach.

Algorithm 1: Iterative solver for the NRPBE.

Input: Initialize the potential $(u^{\mathrm{r}}_{\mathcal{N}}(\mu))^0$, e.g., $(u^{\mathrm{r}}_{\mathcal{N}}(\mu))^0 = 0$ and the tolerance $\delta^0 = 1$.
Output: The converged NRPBE solution $(u^{\mathrm{r}}_{\mathcal{N}}(\mu))^n$ at $\delta^n \leq$ tol.

1: **while** $\delta^n \geq$ tol **do**
2: Solve the linear system (4.8) for $(u^{\mathrm{r}}_{\mathcal{N}}(\mu))^{n+1}$ using AGMG.
3: $\delta^{n+1} \leftarrow \|(u^{\mathrm{r}}_{\mathcal{N}}(\mu))^{n+1} - (u^{\mathrm{r}}_{\mathcal{N}}(\mu))^n\|_2$.
4: $(u^{\mathrm{r}}_{\mathcal{N}}(\mu))^n \leftarrow (u^{\mathrm{r}}_{\mathcal{N}}(\mu))^{n+1}$.
5: **end while**

4.2 The reduced basis method for the NRPBE

The reduced basis method (RBM) is an example of popular projection-based parametrized model order reduction (PMOR) techniques in which the parameter dependence of the PDE solution is exploited by snapshots (high-fidelity solutions) determined over the parameter domain [40]. Their core objective is to construct a parametric reduced order model (ROM) of low dimension, which accurately approximates the original full order model (FOM) or high-fidelity model of high dimension over varying parameter values [41–43]. Other PMOR techniques include proper orthogonal decomposition (POD) [44] and multi-moment matching techniques [45], among others [41].

The RBM leverages an offline/online procedure to ensure an accurate approximation of the high-fidelity solution at extremely low computational costs. It is widely applicable in real-time context such as sensitivity analysis, multimodel simulation, as well as many-query scenarios, e.g., uncertainty quantification and optimal control. For a thorough review of PMOR techniques, see [41].

In the PBE theory, parametric studies are important to investigate different scenarios, for example, to determine protein association rates via the Brownian dynamics simulations [5]. These association rates depend on the ionic strength of the solvent. The RBM is the method of choice to make these simulations faster than doing individual full-order simulations due to the large dimension of the electrostatic potential.

3 **AGMG** implements an aggregation-based algebraic multigrid method, which solves algebraic systems of linear equations, and is expected to be efficient for large systems arising from the discretization of scalar second order elliptic PDEs [39].

It is prohibitively expensive to solve the $\mathcal{N} \times \mathcal{N}$ system in (4.8) for an accurate approximation of $u^{\mathrm{r}}_{\mathcal{N}}(\mu)$ because the dimension $\mathcal{N}$ ranges between $\mathcal{O}(10^6)$ and $\mathcal{O}(10^8)$ for typical biomolecules of interest. Therefore, we exploit the RBM to significantly reduce the computational costs by projecting (via Galerkin) the FOM (4.8) onto a low dimensional subspace (the reduced basis space) which preserves the parametric properties and important system configurations of the FOM. The resultant ROM, of greatly reduced dimension $N \ll \mathcal{N}$, provides an accurate surrogate approximation of $(u^{\mathrm{r}}_{\mathcal{N}}(\mu))^n$, $n = 0, 1, \ldots$.

Algorithm 2: Greedy algorithm.

Input: Training set $\Xi := \{\mu_1, \ldots, \mu_l\} \subset \mathcal{P}$, tolerance $\epsilon_0 = 1$, and potential $(u^{\mathrm{r}}_N(\mu))^0$.
Output: RB basis represented by V and the ROM in (4.13).

1: Choose $\mu^* \in \Xi$ arbitrarily.
2: Solve (3.23) for $u^{\mathrm{r}}_{\mathcal{N}}(\mu^*)$ using Algorithm 1.
3: $V_1 = \left[u^{\mathrm{r}}_{\mathcal{N}}(\mu^*)\right]$, $N = 1$.
4: Orthonormalize V_1.
5: **while** $\max_{\mu \in \Xi} \Delta_N(\mu) \geq \epsilon$ **do**
6: Compute $u^{\mathrm{r}}_N(\mu)$ from (4.13) using Algorithm 4, and calculate $\Delta_N(\mu) = \| r_N(\widehat{u}^{\mathrm{r}}_{\mathcal{N}}(\mu))\|_2$ in (4.12), $\forall\ \mu \in \Xi$.
7: $\mu^* = \arg\max_{\mu \in \Xi} \Delta_N(\mu)$.
8: Solve (3.23) for $u^{\mathrm{r}}_{\mathcal{N}}(\mu^*)$.
9: $V_{N+1} \leftarrow \left[V_N \quad u^{\mathrm{r}}_{\mathcal{N}}(\mu^*)\right]$.
10: Orthonormalize the columns of V_{N+1}.
11: $N \leftarrow N + 1$.
12: **end while**

RBM is based on the assumption that the solution manifold $\mathcal{M}^{\mathcal{N}}$

$$\mathcal{M}^{\mathcal{N}} = \left\{ u^{\mathrm{r}}_{\mathcal{N}}(\mu) : \mu \in \mathcal{P} \right\}, \tag{4.9}$$

is of low dimension. The reduced basis space, which is the space spanned by the snapshots $u^{\mathrm{r}}_{\mathcal{N}}(\mu)$, corresponding to a set of parameter samples, is hierarchically constructed from the solution manifold (4.9), using the greedy procedure summarized in Algorithm 2. The RB space

$$\mathrm{range}(V) = \mathrm{span}\left\{ u^{\mathrm{r}}_{\mathcal{N}}(\mu_1), \ldots, u^{\mathrm{r}}_{\mathcal{N}}(\mu_l) \right\}, \quad \mu_1, \ldots, \mu_l \in \mathcal{P}, \tag{4.10}$$

is nested (hierarchical) in the sense that the previous basis set is a subset of the next until convergence, i.e.,

$$\mathrm{range}(V_1) \subset \mathrm{range}(V_2) \subset \ldots \subset \mathrm{range}(V_N), \tag{4.11}$$

where $N \ll \mathcal{N}$ is the dimension of the ROM. Here, V is the projection matrix which represents the reduced basis space and is obtained from $\mathcal{M}^{\mathcal{N}}$ by the greedy algorithm in Algorithm 2.

The residual in Algorithm 2 is derived from (4.8) and the ROM solution $(\widehat{u}^{\mathrm{r}}_{\mathcal{N}}(\mu))^n = V_N(u^{\mathrm{r}}_N(\mu))^n$ lifted into the high-fidelity space of dimension $\mathcal{N}$, i.e.,

$$r_N((\widehat{u}^{\mathrm{r}}_{\mathcal{N}}(\mu))^{n+1}) = F((\widehat{u}^{\mathrm{r}}_{\mathcal{N}}(\mu))^n) - A((\widehat{u}^{\mathrm{r}}(\mu))^n)(\widehat{u}^{\mathrm{r}}_{\mathcal{N}}(\mu))^{n+1}. \tag{4.12}$$

The ROM for the system (4.8), is therefore, formulated as follows. Given any $\mu \in \mathcal{P}$, and an initial guess $(u^{\mathrm{r}}_N(\mu))^0 \in \mathbb{R}^N$, the RB approximation $(u^{\mathrm{r}}_N(\mu))^{n+1}$, at the future iteration step $n+1$ satisfies the equation

$$A_N((u^{\mathrm{r}}_N(\mu))^n)(u^{\mathrm{r}}_N(\mu))^{n+1} = F_N((u^{\mathrm{r}}_N(\mu))^n), \quad n = 0, 1, \ldots, \tag{4.13}$$

where $(u_N^{\mathrm{r}}(\mu))^0$ is the zero initial guess in this study and A_N and F_N are defined explicitly as

$$A_N := \widehat{A}_1(u_N^{\mathrm{r}}(\mu))^{n+1} + \mu\widehat{A}_2(\widetilde{B}V_N)^n(u_N^{\mathrm{r}}(\mu))^{n+1},$$

and

$$F_N := -\mu\widehat{A}_2 \sinh\odot(\widehat{u}_{\mathcal{N}}^{\mathrm{r}}(\mu))^n + \mu\widehat{A}_2(\widetilde{B}V_N)^n(u_N^{\mathrm{r}}(\mu))^n + b_N^{\mathrm{r}} + V_N^T b_2(\mu),$$

where $\widetilde{B} = \mathrm{diag}(\widetilde{\omega}_1, \widetilde{\omega}_2, \dots, \widetilde{\omega}_{\mathcal{N}})$ and

$$\cosh\odot\widehat{u}_{\mathcal{N}}^{\mathrm{r}}(\mu) =: \widetilde{\omega} = \begin{bmatrix} \widetilde{\omega}_1 \\ \widetilde{\omega}_2 \\ \vdots \\ \widetilde{\omega}_{\mathcal{N}} \end{bmatrix}. \tag{4.14}$$

The resuting ROM is given by

$$\begin{aligned} &\widehat{A}_1(u_N^{\mathrm{r}}(\mu))^{n+1} + \mu\widehat{A}_2(\widetilde{B}V_N)^n(u_N^{\mathrm{r}}(\mu))^{n+1} \\ &\quad = -\mu\widehat{A}_2 \sinh\odot(\widehat{u}_{\mathcal{N}}^{\mathrm{r}}(\mu))^n + \mu\widehat{A}_2(\widetilde{B}V_N)^n(u_N^{\mathrm{r}}(\mu))^n + b_N^{\mathrm{r}} + V_N^T b_2(\mu), \end{aligned} \tag{4.15}$$

where $(u_N^{\mathrm{r}}(\mu))^{n+1}$ is the unknown solution to the ROM.

The reduced matrices $\widehat{A}_1 := V_N^T A_1 V_N$ and $\widehat{A}_2 := V_N^T A_2$ and the reduced vector $b_N^{\mathrm{r}} = V_N^T b_1^{\mathrm{r}}$ (see (4.5)) are determined via projection with the RB basis V_N and can be precomputed in the offline phase of the greedy algorithm. However, the matrix $(\widetilde{B}V_N)^n$ and vector $b_2(\mu)$ are updated and or changed at each iteration and for varying parameter values, respectively, hence, they cannot be precomputed. This leads to a partial offline-online decomposition scenario, whereby Galerkin projections to some terms have to be computed in the online phase.

Note that $V_N^T b_2(\mu)$ in (4.15) is computed by first evaluating a long vector $b_2(\mu)$, then projecting it onto the low dimensional space N using V_N. This is time consuming when $b_2(\mu)$ needs to be evaluated many times for many values of μ. In Section 4.2.1, we propose to apply DEIM to further reduce the computational complexity of $V_N^T b_2^{\mathrm{r}}(\mu)$. Details can be found in [27], where DEIM was applied to PBE problem.

4.2.1 Computational complexity of the regularized reduced order model

It is well known that another key assumption of the RBM, besides the low dimensionality of the solution manifold, is the parameter affine property, which ensures the efficiency of the offline-online decomposition by eliminating the dependency of the ROM on the dimension $\mathcal{N}$ of the truth high-fidelity FOM [40]. However, note that on the one hand, (4.5) is actually parameter nonaffine with respect to the Yukawa-type boundary conditions, represented by F in (4.7). On the other hand, the matrix A_2 requires updates at each iteration; hence Galerkin projections are unavoidable in the online phase.

In this study, we apply DEIM to the parametric nonaffine boundary conditions, the term $b_2(\mu)$. The main idea of DEIM is to significantly reduce the computational complexity of the nonaffine function by interpolation, whereby only a few entries are computed [22]. Before invoking DEIM, snapshots of the nonaffine function $b_2(\mu)$ must be computed for a set of parameter μ in the training set $\Xi = \{\mu_1, \dots, \mu_l\} \subset \mathcal{P}$ and the snapshot matrix,

$$G = [b_2(\mu_1), \dots, b_2(\mu_l)] \in \mathbb{R}^{\mathcal{N}\times l}, \tag{4.16}$$

is constructed.

Then, the singular value decomposition (SVD) of G is computed,

$$G = U_G \Sigma W^T, \tag{4.17}$$

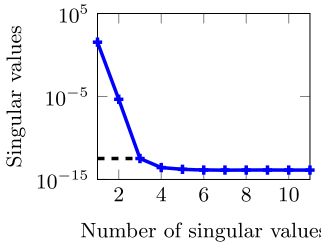


Figure 3: Decay of singular values of Σ with number of singular values in (4.17).

where $U_G \in \mathbb{R}^{\mathcal{N}\times l}$, $\Sigma \in \mathbb{R}^{l\times l}$, and $W \in \mathbb{R}^{l\times l}$. Note that the matrices U_G and W are orthogonal, i.e., $(U_G)^T U_G = W^T W = I_l$, $I_l \in \mathbb{R}^{l\times l}$ and $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_l)$, with $\sigma_1 \geq \dots \geq \sigma_l \geq 0$. Figure 3 shows the decay of singular values of Σ for the protein Fasciculin 1. We truncate the singular values being smaller than 10^{-13} and retain $r = 3$ singular values.

DEIM seeks to approximate the function $b_2(\mu)$ with the linear combination of the basis vectors $U_G = \left[u_1^G, \dots, u_r^G\right] \in \mathbb{R}^{\mathcal{N}\times r}$, i.e.

$$b_2(\mu) \approx U_G c(\mu), \tag{4.18}$$

where $c(\mu) \in \mathbb{R}^r$ is the corresponding coefficient vector, which is determined under the assumption that $U_G c(\mu)$ interpolates $b_2(\mu)$ at r selected interpolation points, then,

$$P^T b_2(\mu) = P^T U_G c(\mu), \tag{4.19}$$

where P is an index matrix given by

$$P = [e_{\wp_1}, \dots, e_{\wp_r}] \in \mathbb{R}^{\mathcal{N}\times r}, \tag{4.20}$$

which consists of unit vectors $e_{\wp_i}$, $i = 1, \dots, r$. Here, the indices $\wp_i$, are the DEIM interpolation points which are selected iteratively with the greedy iteration as presented in Algorithm 3.

Algorithm 3: DEIM algorithm [22, 46].

Input: POD basis $\left\{u_i^G\right\}_{i=1}^r$ for G in Eq. (4.17)).
Output: DEIM basis U_G and indices $\vec{\wp} = [\wp_1, \dots, \wp_r]^T \in \mathbb{R}^r$.

1: $\wp_1 = \arg\max_{j\in\{1,\dots,\mathcal{N}\}} |u_{1j}^G|$, where $u_1^G = \left(u_{11}^G, \dots, u_{1\mathcal{N}}^G\right)^T$.
2: $U_G = \left[u_1^G\right]$, $P = [e_{\wp_1}]$, $\vec{\wp} = [\wp_1]$.
3: **for** $i = 2$ to r **do**
4: Solve $\left(P^T U_G\right)\alpha = P^T u_i^G$ for α, where $\alpha = (\alpha_1, \dots, \alpha_{i-1})^T$.
5: $r_i = u_i^G - U_G\alpha$.
6: $\wp_i = \arg\max_{j\in\{1,\dots,\mathcal{N}\}} |r_{ij}|$, where $r_i = (r_{i1}, \dots, r_{i\mathcal{N}})^T$.
7: $U_G \leftarrow \left[U_G\ u_i^G\right]$, $P \leftarrow [P\ e_{\wp_i}]$, $\vec{\wp} \leftarrow \begin{bmatrix}\vec{\wp}\\ \wp_i\end{bmatrix}$.
8: **end for**

Suppose that $P^T U_G \in \mathbb{R}^{r\times r}$ is nonsingular, then $c(\mu)$ can be determined from the overdetermined system (4.19) by

$$c(\mu) = \left(P^T U_G\right)^{-1} P^T b_2(\mu). \tag{4.21}$$

Therefore, the function $b_2(\mu)$ in [9] can be approximated as

$$b_2(\mu) \approx U_G c(\mu) = U_G \left(P^T U_G\right)^{-1} P^T b_2(\mu). \tag{4.22}$$

The ROM in (4.15) with DEIM approximation becomes

$$\widehat{A}_1(u_N^{\mathrm{r}}(\mu))^{n+1} + \mu\widehat{A}_2(\widetilde{B}V_N)^n(u_N^{\mathrm{r}}(\mu))^{n+1} = -\mu\widehat{A}_2 \ \sinh\odot(\widehat{u}_{\mathcal{N}}^{\mathrm{r}}(\mu))^n + \mu\widehat{A}_2(\widetilde{B}V_N)^n(u_N^{\mathrm{r}}(\mu))^n + b_N^{\mathrm{r}}(\mu) + V_N^T U_G \left(P^T U_G\right)^{-1} P^T b_2(\mu). \tag{4.23}$$

The interpolant $V_N^T U_G \left(P^T U_G\right)^{-1} P^T b_2(\mu)$ can be computed a lot cheaper than $V_N^T b_2(\mu)$ because $V_N^T U_G \left(P^T U_G\right)^{-1}$ can be precomputed independently of the parameter μ. Only those entries in $b_2(\mu)$ that correspond to the interpolation indices $\wp_i, i = 1, \ldots, r$, $r \ll \mathcal{N}$, i.e., $P^T b_2(\mu)$ can be computed instead of the entire $\mathcal{N}$ entries in $b_2(\mu)$. This saves significant computational efforts when $b_2(\mu)$ needs to be repeatedly computed for different values of μ.

Note that at each iteration only a small ROM in (4.23) is solved. With its small size $N \ll \mathcal{N}$, the system (4.23) can be solved using a direct solver rather than the iterative solver (AGMG), which is applied to the FOM in (4.3). The iterative approach of obtaining an approximate solution $V_N(u_N^{\mathrm{r}}(\mu))^{n+1}$ to (4.1) using the ROM (4.23) is summarized in Algorithm 4.

Algorithm 4: Iterative solver for the regularized ROM in (4.23).

Input: Initialize the potential $(u_N^{\mathrm{r}}(\mu))^0$, e.g., $(u_N^{\mathrm{r}}(\mu))^0 = 0$, tolerance tol >0, and $\delta^0 = 1$.
Output: The converged ROM solution $(u_N^{\mathrm{r}}(\mu))^n$ at $\delta^n \leq$ tol.

1: Precompute $\widehat{A}_1$, b_N^{r} in (4.23) and U_G and $\vec{\wp}$ in Algorithm 3.
2: **while** $\delta^n \geq$ tol **do**
3: Assemble the ROM in (4.23) using the precomputed quantities in Step 1.
4: Solve the regularized ROM (4.23) for $(u_N^{\mathrm{r}}(\mu))^{n+1}$.
5: $\delta^{n+1} \leftarrow \|(u_N^{\mathrm{r}}(\mu))^{n+1} - (u_N^{\mathrm{r}}(\mu))^n\|_2$.
6: $(u_N^{\mathrm{r}}(\mu))^n \leftarrow (u_N^{\mathrm{r}}(\mu))^{n+1}$.
7: **end while**

Remark 5. The total electrostatic potential is obtained by lifting the reduced order long-range surrogate solution into the high-fidelity space $\mathcal{N}$ and adding to the parameter independent analytically precomputed short-range component $\mathbf{P}_{\mathrm{s}}$ in (3.14), i.e.,

$$u(\mu) = \mathbf{P}_{\mathrm{s}} + \widehat{u}_{\mathcal{N}}^{\mathrm{r}}(\mu), \tag{4.24}$$

where $\widehat{u}_{\mathcal{N}}^{\mathrm{r}}(\mu) = V_N u_N^{\mathrm{r}}(\mu)$.

4.3 The reduced basis method for the classical NPBE

In this section, we apply RBM to the classical nonlinear PBE (NPBE), and compare this version in Section 5 with the suggested approach from Section 4.2. We begin by considering the FOM of the classical NPBE in (2.1) after discretization in space, i.e.,

$$A(u^{\mathcal{N}}(\mu)) = f(\mu), \quad \mu \in \mathcal{P}, \tag{4.25}$$

where $f(\mu)$ includes both the singular sources from the right-hand side of (2.1) and the parameter non-affine Dirichlet boundary conditions from (2.2). The corresponding classical ROM is defined as

$$\widehat{A}(u_N(\mu)) = \widehat{f}(\mu), \tag{4.26}$$

where $\widehat{A}(u_N(\mu)) = V_N^T A(V_N u_N(\mu))$ and $\widehat{f} = V_N^T f$. Here, V_N can be constructed using the greedy algorithm in Algorithm 2 by replacing the snapshots in Step 3 and Step 9 with the solutions to (4.25).

Note that the FOM of the classical NPBE is solved iteratively in a similar way like the NRPBE using Algorithm 1. The corresponding iterative form of (4.25) is given by

$$A_1(u^{\mathcal{N}}(\mu))^{n+1} + \mu A_2 B_2^n(u^{\mathcal{N}}(\mu))^{n+1} = -\mu A_2 \ \sinh\odot(u^{\mathcal{N}}(\mu))^n + \mu A_2 B_2^n(u^{\mathcal{N}}(\mu))^n + f(\mu), \tag{4.27}$$

where all the quantities except B_2, f and the solution $(u^{\mathcal{N}}(\mu))^{n+1}$ are equivalent to those in the NRPBE (4.5). Here B_2 is defined as

$$B_2 = \text{diag}(v_1, v_2, \dots, v_{\mathcal{N}}),$$

and is constructed from

$$\cosh \odot u^{\mathcal{N}}(\mu) =: v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{\mathcal{N}} \end{bmatrix}. \tag{4.28}$$

The ROM of (4.27) is straightforward, i.e., given any $\mu \in \mathcal{P}$, and an initial potential distribution $(u_N(\mu))^0$, the RB approximation $(u_N(\mu))^{n+1}$, at the subsequent iteration steps $n+1$ satisfies

$$\widehat{A}_1(u_N(\mu))^{n+1} + \mu\widehat{A}_2(\widetilde{B}_2 V_N)^n(u_N(\mu))^{n+1} = -\mu\widehat{A}_2 \ \sinh \odot(\widehat{u}^{\mathcal{N}}(\mu))^n + \mu\widehat{A}_2(\widetilde{B}_2 V_N)^n(u_N(\mu))^n + \widehat{f}(\mu), \tag{4.29}$$

where $(\widehat{u}^{\mathcal{N}}(\mu))^n = V_N(u_N(\mu))^n$ and $\widetilde{B}_2 = \text{diag}(\widetilde{v}_1, \widetilde{v}_2, \dots, \widetilde{v}_{\mathcal{N}})$ is constructed from

$$\cosh \odot \widehat{u}^{\mathcal{N}}(\mu) =: \widetilde{v} = \begin{bmatrix} \widetilde{v}_1 \\ \widetilde{v}_2 \\ \vdots \\ \widetilde{v}_{\mathcal{N}} \end{bmatrix}. \tag{4.30}$$

The process of iteratively solving (4.29) is similar to that of (4.13), which is provided in Algorithm 4.

5 Numerical results

Consider $n^{\otimes 3}$ 3D uniform Cartesian grids, in a cubic domain $[a, b]^3$, for computing the reduced basis approximation of the NRPBE on a modest PC which possesses the following specifications: Intel (R) Core (TM) *i*7−4790 CPU @ 3.60 GHz with 8 GB RAM. In this study, the NRPBE is discretized by the finite difference method (FDM) to obtain the FOM and the numerical computations are implemented in MATLAB, version R2017b.

In the numerical tests, the molecular charge density function (singular source term) for the classical NPBE and the regularized Dirac density function for the NRPBE are obtained from PQR[4] files which are generated from the following biomolecules with varying sizes that depend on the number of atoms:

(a) The acetazolamide molecule consisting of 18 atoms, which is used as a ligand in the human carbonic anhydrase (hca) protein−ligand complex for the calculation of the binding energy [47, 48].
(b) Fasciculin 1, an anti-acetylcholinesterase toxin from green mamba snake venom [49] consisting of 1228 atoms.
(c) A 180-residue cytokine solution NMR structure of a murine-human chimera of leukemia inhibitory factor (LIF) [50] consisting of 2809 atoms.

Remark 6. Since the solution of the PBE has a slow polynomial decay in $1/\|\bar{x}\|$, it is paramount that large domains, approximately 3-times the size of the biomolecule be used in order to accurately approximate the boundary conditions [8]. In this regard, we use domains of lengths $(32\ \text{Å})^3$, $(60\ \text{Å})^3$, and $(65\ \text{Å})^3$, respectively, for the aforementioned biomolecules. Here, Å denotes the angstrom unit of length.

4 A PQR (or position, charge (Q), and radius) file is a protein data bank (PDB) file with the temperature and occupancy columns replaced by columns containing the per-atom charge (Q) and radius (R) using the pdb2pqr software. PQR files are used in several computational biology packages, including APBS [53].

To begin with, we demonstrate the solution components of the full order model (FOM) of the NRPBE via the RS tensor format for the protein Fasciculin 1 in case (b), in a uniform Cartesian grid of $129^{\otimes 3}$ and a 60 Å domain length. Figure 4 shows the short- and long-range components of the target electrostatic potential, which are computed analytically from the CCT tensor (3.14), and numerically via the NRPBE in (3.23), respectively, and the corresponding total electrostatic potential. Notice that the molecular boundary (interface) is irregular and it encloses the short-range potential component which is represented by the sharp cusps in the figure.

The behaviour in the scaling among the three electrostatic potentials in Figure 4 shows that the total potential on the right-hand side inherits the largest potential value of 0.5 from the short-range component (top left), while the smallest value of $\mathcal{O}(10^{-5})$ is obtained from the long-range component (bottom left) towards the boundary of the domain.

Remark 7. The main computational advantage of applying the RBM technique to the NRPBE is that the RB approximation is only applied to the smooth long-range component of the potential, see Figure 4 (bottom left), thereby avoiding the singularities inherent in the short-range component that are known to cause numerical difficulties. Hence the resultant RB approximation is expected to be of higher accuracy.

5.1 Reduced basis results for the NRPBE

Here, we determine the accuracy and computational efficacy of the RBM approximating the high-fidelity solution to the NRPBE for biomolecular modeling. We set the solute and solvent dielectric coefficients as $\epsilon_{\mathrm{m}} = 2$ and $\epsilon_{\mathrm{s}} = 78.54$, respectively, and employ the parameter values from the training set $\Xi \in \mathcal{P} = [0.05, 0.15]$ with a sample size of $l = 11$, greedy tolerance $\mathrm{tol} = 10^{-10}$ for Algorithm 2 to generate the projection matrix V_N. Furthermore the residual in (4.12) is used as an error estimator for the ROM in the greedy algorithm in Algorithm 2.

First, we consider the NRPBE system generated by all the three cases (a), (b), and (c), in uniform Cartesian grids of $97^{\otimes 3}$ for case (a) and $129^{\otimes 3}$ for cases (b) and (c), respectively. We show in Table 1, the decay of the

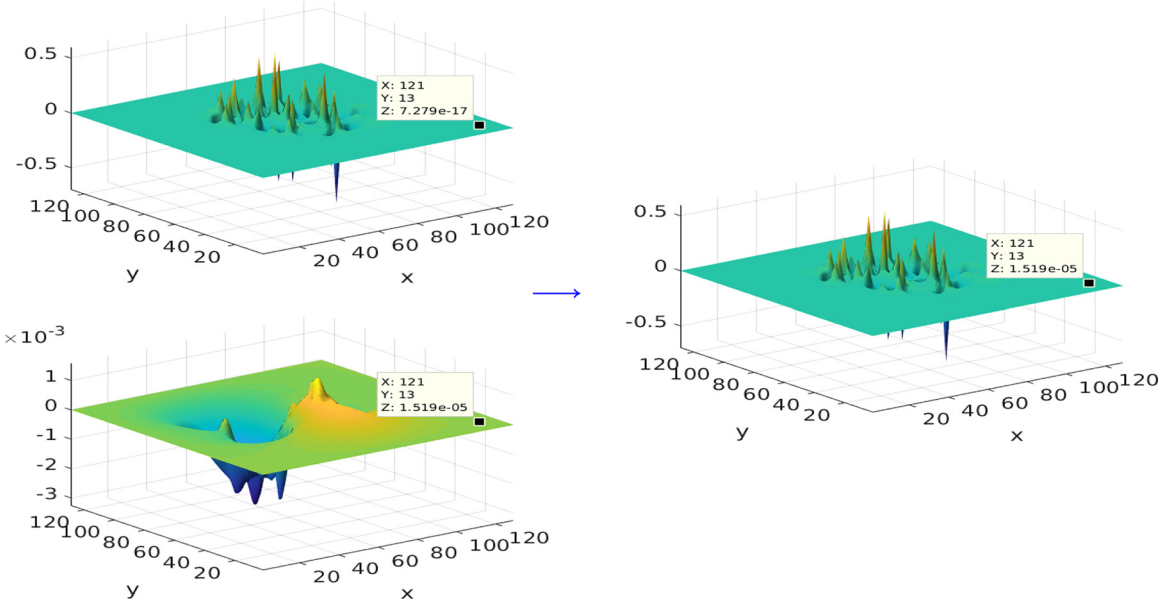


Figure 4: Short-range (top left) and long-range (bottom left) contributions to the total electrostatic potential (right) of the Fasciculin 1 protein in solution.

Table 1: The comparison between the maximal error estimator $\Delta_N^{\max}(\mu)$ and the true error for the NRPBE during the greedy iteration at the current RB dimension $i = 1, \ldots, N$ for the biomolecules in cases (a) to (c).

System	Error at iteration 1		Error at iteration 2		ROM dimension N
	$\Delta_N^{\max}(\mu)$	True error	$\Delta_N^{\max}(\mu)$	True error	
Acetazolamide	5.0573×10^{-6}	1.2719×10^{-8}	3.0339×10^{-12}	3.0395×10^{-15}	2
Fasciculin-1	1.0685×10^{-5}	8.9228×10^{-8}	3.6895×10^{-12}	2.0232×10^{-14}	2
LIF	3.2610×10^{-5}	1.4510×10^{-7}	2.0573×10^{-11}	3.2015×10^{-14}	2

maximal error estimator, defined as

$$\Delta_N^{\max}(\mu) = \max_{\mu \in \mathcal{P}} \| r_N(\widehat{u}_N^{\mathrm{r}}; \mu) \|_2,$$

and the true error $\| u_{\mathcal{N}}^{\mathrm{r}}(\mu) - \widehat{u}_N^{\mathrm{r}}(\mu) \|_2$, during the greedy algorithm at the current RB dimension $i = 1, \ldots, N$ for all of these cases.

Note that the ROM provides highly accurate approximations, close to machine precision ($\mathcal{O}(10^{-15})$) for the NRPBE as demonstrated by the true error in the second iteration. This is due to the smoothness of the long-range electrostatic potential, which enhances rapid and accurate model reduction process and facilitates, in general, low-rank approximation.

Next, we validate the final ROM at 100 random $\mu \in \mathcal{P}$ in Figure 5. It is clear that the true error of the ROM is still below the tolerance for all 100 $\mu \in \mathcal{P}$.

5.2 Comparison of the RB approximation accuracy between the NRPBE and the NPBE

In this section, we demonstrate via the RB approximation, that the NRPBE model is more accurate and computationally efficient than the classical NPBE. In a similar style as in Section 5.1, we consider the biomolecules in cases (a) to (c) with the corresponding domain lengths and grid dimensions. We demonstrate the accuracy of the RB approximation for the classical NPBE model in order to compare it with the NRPBE model. We begin by demonstrating in Figure 6, the comparison of the error decay between the maximal error estimator $\Delta_N^{\max}(\mu)$ and the true error for the classical NPBE during the greedy iteration at the current RB dimension $i = 1, \ldots, N$ for the biomolecules in cases (a) to (c).

We notice that the RBM constructs a small ROM (i.e., $N = 2$) of high accuracy ($\mathcal{O}(10^{-12})$) for the NRPBE in Table 1 because of the regularized nature of the model, the RBM applied to the classical NPBE, nevertheless, generates a ROM of dimension $N = 6$ at the accuracy of $\mathcal{O}(10^{-4})$ for most biomolecules in Figure 6 [27, 51]. This

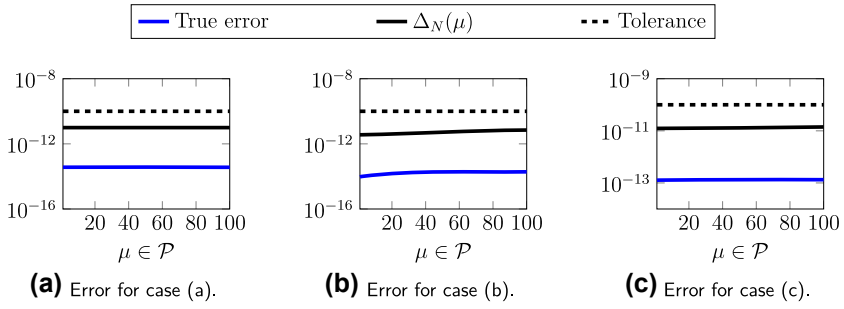
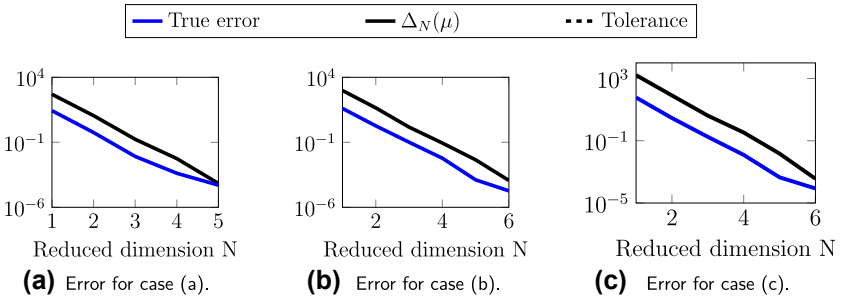


Figure 5: Comparison between the maximal error estimator and the true error for the NRPBE for acetazolamide, fasciculin 1, and LIF systems for the final ROM at 100 random (varying) parameter values $\mu \in \mathcal{P}$. Here, tolerance is the allowed maximum error between the error estimator and the true error. Henceforth, we have error estimator (solid black line), true error (solid blue line), and tolerance (dashed black line).



(a) Error for case (a). **(b)** Error for case (b). **(c)** Error for case (c).

Figure 6: The comparison between the maximal error estimator and the true error for the classical NPBE during the greedy iteration at the current RB dimension $i = 1, \ldots, N$ for the biomolecules for acetazolamide, fasciculin 1, and LIF systems.

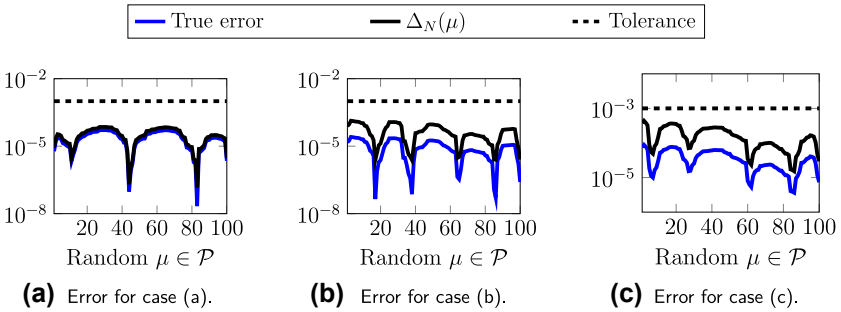
is because in the latter case, the short-range component of the electrostatic potential impedes the reduction process due to the sharp cusps or singularities which are hard to capture in the ROM. Furthermore, case (a) has a slightly smaller ROM dimension due to its small number of atoms as compared to the rest, hence its small number of solution singularities (cusps) to be captured in the ROM.

The accuracy of the RB approximation of the classical NPBE in Figure 7 is much lower than that of the NRPBE in Figure 5 due to the inaccurate approximation of the short-range component inherent in the former. This demonstrates the efficacy of the regularization scheme based on the RS tensor technique. The oscillations in the error in Figure 7 also justifies the irregularity of the singular solution, which impedes the model reduction process.

5.2.1 Runtimes and computational speed-ups

We compare the computational runtime of computing both the classical and regularized NPBE models as well as that of the corresponding ROM (using the RBM) in Table 2. The respective PBE models were applied to the protein fasciculin 1. Given a fixed value of the parameter μ, Table 2 compares the runtimes for solving the FOM (using the FDM), constructing the ROM (using the RBM), and solving the ROM (using direct methods), for the classical LPBE and NPBE with those of the regularized LRPBE and NRPBE, respectively. It is clear that the RBM spends more time in the offline phase of the greedy algorithm to compute snapshots for the classical NPBE than on the NRPBE model, see Figure 6 and Table 1. This is mainly because of the presence of rapid singularities in the PBE solution, which provides an onerous task in the construction of the ROM.

Consequently, Table 3, shows that solving the FOM for the NRPBE by the classical numerical techniques (in this case, the FDM) is sufficient and computationally efficient only for a single parameter value. However, for many varying parameter values, the RBM is more efficient because it constructs only a small ROM once, which can then be solved fast to obtain the solutions at any values of the parameter. For instance, for 1000 different parameter values, the ROM runtime is $\approx 6.59 \times 10^{-3}$ sec, leading to a total runtime of ≈ 82.97 sec to solve the NRPBE using the RBM technique instead of ≈ 28300 sec by the FDM solver. Note that the runtimes



(a) Error for case (a). **(b)** Error for case (b). **(c)** Error for case (c).

Figure 7: Comparison between the error estimator and the true error for the classical NPBE for acetazolamide, fasciculin 1, and LIF systems for the final ROM at 100 random (varying) parameter values $\mu \in \mathcal{P}$.

Table 2: Runtimes for the FOM, RBM, and ROM for the linear and nonlinear variants of both the classical and the regularized PBE models.

	Runtime (seconds) for the PBE and the RPBE			
	LPBE	**NPBE**	**LRPBE**	**NRPBE**
FOM	17.68	34.40	22.83	28.30
RBM	107.98	238.78	90.04	76.38
ROM	2.22×10^{-2}	2.40×10^{-2}	2.10×10^{-3}	6.59×10^{-3}

Table 3: Comparison of the runtimes and speed-ups between the FOM and the RBM for the NRPBE in a multiparameter context.

	Runtime (seconds) and speed-up using the FDM and the RBM		
No. of parameters	**FOM for NRPBE**	**RBM for NRPBE**	**Speed-up**
1	28.30	76.38	0.37
10	≈283.00	≈76.44	3.70
100	≈2830.00	≈77.04	36.73
1000	≈28300.00	≈82.97	341.09

for the 1000 varying parameter values for the FOM are mere approximations based on that of the single parameter value, since simulating the FOM for so many times is impractical.

6 Conclusions

In this study, we review the salient properties of the RS canonical tensor format as a regularization scheme for the nonlinear PBE (NPBE) for calculating the electrostatic potential within and around biomolecules as proposed in [18, 19]. Among these properties is the grid-based RS tensor splitting of the Dirac delta distribution into the smooth and singular source term components. The NPBE is then discretized with the smooth approximation of the Dirac delta distribution, yielding a regularized FOM that is devoid of the singularities in the resultant solution. The RBM leverages this property by constructing a regularized ROM at extremely low computational costs as compared to that of the classical variant. This avoids constructing a ROM which comprises of the highly singular component of the electrostatic potential, thereby reducing the errors in the numerical approximation. The total potential is obtained by adding the regularized component (solution of the ROM), which is lifted (by projection) to the high-fidelity space, $\mathcal{N}$, to the directly precomputed canonical tensor representation of the short-range component of the Newton kernel.

Acknowledgement: The authors thank the following organizations for financial and material support on this project: International Max Planck Research School (IMPRS) for Advanced Methods in Process and Systems Engineering and Max Planck Society for the Advancement of Science (MPG).
Author contribution: All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.
Research funding: None declared.
Conflict of interest statement: The authors declare no conflicts of interest regarding this article.

References

[1] N. A. Baker, "Poisson-Boltzmann methods for biomolecular electrostatics," *Methods Enzymol.*, vol. 383, pp. 94–118, 2004.

[2] N. A. Baker, "Biomolecular applications of Poisson–Boltzmann equation," in *Reviews in Computational Chemistry*, vol. 21, Hoboken, NJ, USA, John Wiley & Sons, 2005.

[3] N. A. Baker and J. Wagoner, "Solvation forces on biomolecular structures: a comparison of explicit solvent and Poisson-Boltzmann models," *J. Comput. Chem.*, vol. 25, pp. 1623–1629, 2004.

[4] B. Z. Lu, Y. C. Zhou, M. J. Holst, and J. A. McCammon, "Recent progress in numerical methods for Poisson-Boltzmann equation in biophysical applications," *Commun. Comput. Phys.*, vol. 3, no. 5, pp. 973–1009, 2008.

[5] F. Fogolari, A. Brigo, and H. Molinari, "The Poisson-Boltzmann equation for biomolecular electrostatics: a tool for structural biology," *J. Mol. Recogn.*, vol. 15, no. 6, pp. 377–392, 2002.

[6] M. T. Neves-Petersen and S. Petersen, "Protein electrostatics: a review of the equations and methods used to model electrostatic equations in biomolecules - applications in biotechnology," *Biotechnol. Annu. Rev.*, vol. 9, pp. 315–395, 2003.

[7] M. Stein, R. R. Gabdoulline, and R. C. Wade, "Cross-species analysis of the glycoliticmpathway by comparison of molecular interaction fields," *Mol. Biosyst.*, vol. 6, pp. 162–174, 2010.

[8] M. J. Holst, "Multilevel methods for the Poisson-Boltzmann equation," *Ph.D. thesis*, Urbana-Champaign, IL, USA, Numerical Computing Group, University of Illinois, 1994.

[9] F. Dong, B. Oslen, and N. A. Baker, "Computational methods for biomolecular electrostatics," *Methods Cell Biol.*, vol. 84, no. 1, pp. 843–870, 2008.

[10] N. A. Baker, M. J. Holst, and F. Wang, "The adaptive multilevel finite element solution of the Poisson-Boltzmann equation on massively parallel computers," *IBM J. Res. Dev.*, vol. 45, pp. 427–438, 2001.

[11] J. Wang and R. Luo, "Assessment of linear finite difference Poisson-Boltzmann solvers," *J. Comput. Chem.*, vol. 31, pp. 1689–1698, 2010.

[12] M. Holst, N. Baker, and F. Wang, "Adaptive multilevel finite element solution of the Poisson-Boltzmann equation: algorithms and examples," *J. Comput. Chem.*, vol. 21, pp. 1319–1342, 2000.

[13] A. H. Boschitsch and M. O. Fenley, "Hybrid boundary element and finite difference method for solving the nonlinear Poisson-Boltzman equation," *J. Comput. Chem.*, vol. 25, no. 7, pp. 935–955, 2004.

[14] H. X. Zhou, "Boundary element solution of macromolecular electrostatics: inteaction energy between two proteins," *Biophys. J.*, vol. 65, no. 2, pp. 955–963, 1993.

[15] D. Xie, "New solution decomposition and minimization scheme for Poisson-Boltzmann equation in calculation of biomolecular electrostatics," *J. Comput. Phys.*, vol. 275, pp. 294–309, 2014.

[16] L. Chen, M. J. Holst, and J. Xu, "The finite element approximation of the nonlinear Poisson-Boltzmann equation," *SIAM J. Numer. Anal.*, vol. 45, no. 6, pp. 2298–2320, 2007.

[17] M. Mirzadeh, M. Theillard, A. Helgadottir, D. Boy, and F. Gibou, "An adaptive, finite difference solver for the nonlinear Poisson-Boltzmann equation with applications to biomolecular computations," *Commun. Comput. Phys.*, vol. 13, no. 1, pp. 150–173, 2013.

[18] P. Benner, V. Khoromskaia, B. Khoromskij, C. Kweyu, and M. Stein, "Regularization of Poisson-Boltzmann type equations with singular source terms using the range-separated tensor format," *SIAM J. Sci. Comput.*, vol. 43, no. 1, pp. A415–A445, 2021.

[19] C. Kweyu, V. Khoromskaia, B. Khoromskij, M. Stein, and P. Benner, "Solution decomposition for the nonlinear Poisson-Boltzmann equation using the range-separated tensor format," arXiv:2109.14073, 2021.

[20] P. Benner, V. Khoromskaia, and B. N. Khoromskij, "Range-separated tensor format for many-particle modeling," *SIAM J. Sci. Comput.*, vol. 40, no. 2, pp. A1034–A1062, 2018.

[21] L. Ji, Y. Chen, and Z. Xu, "A reduced basis method for the nonlinear Poisson-Boltzmann equation," *Adv. Appl. Math. Mech.*, vol. 11, pp. 1200–1218, 2019.

[22] S. Chaturantabut and D. C. Sorensen, "Nonlinear model reduction via discrete empirical interpolation," *SIAM J. Sci. Comput.*, vol. 32, no. 5, pp. 2737–2764, 2010.

[23] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera, "Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations," *ESAIM Math. Model. Numer. Anal.*, vol. 41, no. 3, pp. 575–605, 2007.

[24] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, "An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations," *C. R. Math. Acad. Sci. Paris*, vol. 339, no. 9, pp. 667–672, 2004.

[25] B. N. Khoromskij, "Range-separated tensor representation of the discretized multidimensional Dirac delta and elliptic operator inverse," *J. Comput. Phys.*, vol. 401, p. 108998, 2020.

[26] K. A. Sharp and B. Honig, "Electrostatic interactions in macromolecules: theory and applications," *Annu. Rev. Biophys. Chem.*, vol. 19, pp. 301–332, 1990.

[27] C. Kweyu, L. Feng, M. Stein, and P. Benner, "Fast solution of the Poisson-Boltzmann equation with nonaffine parametrized boundary conditions using the reduced basis method," *Comput. Visual Sci.*, vol. 23, p. 15, 2020.
[28] F. Fogolari, P. Zuccato, G. Esposito, and P. Viglino, "Biomolecular electrostatics with the linearized Poisson-Boltzmann equation," *Biophys. J.*, vol. 76, no. 1, pp. 1–16, 1999.
[29] C. Qin, H. Meng-Juei, W. Jun, and L. Ray, "Performance of nonlinear finite-difference Poisson-Boltzmann solvers," *J. Chem. Theor. Comput.*, vol. 6, no. 1, pp. 203–211, 2010.
[30] I. Chern, J. Liu, and W. Wang, "Accurate evaluation of electrostatics for macromolecules in solution," *Methods Appl. Anal.*, vol. 10, no. 2, pp. 309–328, 2003.
[31] Encyclopedia of Mathematics, *Newton potential*, 2018. Available at: http://www.encyclopediaofmath.org/index.php?title=Newton_potential&oldid=33114 [accessed: 12 03, 2018].
[32] V. Khoromskaia and B. N. Khoromskij, "Grid-based lattice summation of electrostatic potentials by assembled rank-structured tensor approximation," *Comput. Phys. Commun.*, vol. 185, no. 12, pp. 3162–3174, 2014.
[33] B. N. Khoromskij and V. Khoromskaia, "Multigrid accelerated tensor approximation of function related multidimensional arrays," *SIAM J. Sci. Comput.*, vol. 31, no. 4, pp. 3002–3026, 2009.
[34] W. Rocchia, E. Alexov, and B. Honig, "Extending the applicability of the nonlinear Poisson-Boltzmann equation: multiple dielectric constants and multivalent ions," *J. Phys. Chem.*, vol. 105, no. 28, pp. 6507–6514, 2001.
[35] B. A. Luty, M. E. Davis, and J. A. McCammon, "Solving the finite-difference nonlinear Poisson-Boltzmann equation," *J. Comput. Chem.*, vol. 13, no. 9, pp. 1114–1118, 1992.
[36] H. Oberoi and N. M. Allewell, "Multigrid solution of the nonlinear Poisson-Boltzmann equation and calculation of titration curves," *Biophys. J.*, vol. 65, no. 1, pp. 48–55, 1993.
[37] M. Holst and F. Saied, "Numerical solution of the nonlinear Poisson-Boltzmann equation: developing more robust and efficient methods," *J. Comput. Chem.*, vol. 16, pp. 337–364, 1995.
[38] A. I. Shestakov, J. L. Milovich, and A. Noy, "Solution of the nonlinear Poisson-Boltzmann equation using pseudo-transient continuation and the finite element method," *Commun. Comput. Phys.*, vol. 247, pp. 62–79, 2002.
[39] Y. Notay, "An aggregation-based algebraic multigrid method," *Electron. Trans. Numer. Anal.*, vol. 37, pp. 123–146, 2010.
[40] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, Switzerland, Springer International Publishing, 2016.
[41] P. Benner, S. Gugercin, and K. Willcox, "A survey of model reduction methods for parametric systems," *SIAM Rev.*, vol. 57, no. 4, pp. 483–531, 2015.
[42] J. L. Eftang, "Reduced basis methods for parametrized partial differential equations," Ph.D. thesis, Trondheim, Norway, Norwegian University of Science and Technology, 2011.
[43] G. Rozza, D. B. P. Huynh, and A. T. Patera, "Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations," *Arch. Comput. Methods Eng.*, vol. 15, no. 3, pp. 229–275, 2008.
[44] S. Volkwein, *Model Reduction Using Proper Orthogonal Decomposition. Lecture Notes*, Konstanz, University of Konstanz, 2013.
[45] L. Feng and P. Benner, "A robust algorithm for parametric model order reduction based on implicit moment matching," in *Reduced Order Methods for Modeling and Computational Reduction, MS&A Series*, vol. 9, Berlin, Heidelberg, New York, Springer-Verlag, 2014, pp. 159–186. chapter 6.
[46] L. Feng, M. Mangold, and P. Benner, "Adaptive POD-DEIM basis construction and its application to a nonlinear population balance system," *AIChE J.*, pp. 3832–3844, 2017, https://doi.org/10.1002/aic.15749.
[47] M. Holst and F. Saied, "Multigrid solution of the Poisson-Boltzmann equation," *J. Comput. Chem.*, vol. 14, pp. 105–113, 1993.
[48] S. Vergara-Perez and M. Marucho, "MPBEC, a Matlab program for biomolecular electrostatic calculations," *Comput. Phys. Commun.*, vol. 198, pp. 179–194, 2016.
[49] M. H. le Du, P. Marchot, P. E. Bougis, and J. C. Fontecilla-Camps, "1.9 Angstrom resolution structure of fasciculine 1, an anti-acetylcholinesterase toxin from green mamba snake venom," *J. Biol. Chem.*, vol. 267, pp. 22122–22130, 1992.
[50] M. G. Hinds, T. Maurer, J. Zhang, and N. A. Nicola, "Solution structure of Leukemia inhibitory factor," *Biol. Chem.*, vol. 273, pp. 13738–13745, 1998.
[51] C. Kweyu, M. Hess, L. Feng, M. Stein, and P. Benner, "Reduced basis method for Poisson-Boltzmann Equation," in *ECCOMAS Congress 2016 - Proc. of the VII European Congress on Computational Methods in Applied Sciences and Engineering*, vol. 2, M. Papadrakakis, V. Papadopoulos, G. Stefanou, and V. Plevris, Eds., Athens, National Technical University of Athens, 2016, pp. 4187–4195.
[52] J. O. Bockris and A. K. N. Reddy, *Modern Electrochemistry: Ionics*, New York, Plenum Press, 1998.
[53] N. A. Baker, D. Sept, S. Joseph, M. J. Holst, and J. A. McCammon, "Electrostatics of nanosystems: application to microtubules and the ribosome," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 98, no. 18, pp. 10037–10041, 2001.