B-SPLINE WAVELET INTEGRAL TENSORIZATION ON NURBS MANIFOLDS

MAHARAVO RANDRIANARIVONY

Abstract. The Boundary Element Method originating from the Poisson-Boltzmann equation is considered for the ionic interaction between solute and solvent media that are separated by a molecular surface decomposed into four-sided patches. We present a method which allows to simultaneously compute many wavelet integrals admitting patches as domains of integration. The presented approach based on higher order tensors is developed for arbitrary dimensions because some involved integrals are $4D$ and others are $2D$ by using the wavelet-Galerkin setting. Based on higher order singular value decompositions, the integrals in high dimension are reduced to $1D$-integrals which are efficiently computable and which are stored for future look-up tables. Since the molecular surface is decomposed into smooth four-sided patches, the involved integrands are transformed into multivariate functions defined on the hypercube.

1. Introduction

The Poisson-Boltzmann Equation (PBE) is frequently encountered in different areas including plasma physics, ionic solution and highly charged macro-particles. Wavelets [1, 2, 3] have gained considerable attentions in the domain of integral equations because they significantly compress a dense matrix into a quasi-sparse one. Only very few matrix entries are relevant as examined by many authors including [4, 5]. The main difficulty for the wavelet based integral equation is the evaluation of the matrix entries which are typically $4D$ or $2D$ integrals admitting integrands that are highly nonlinear and partially singular. We propose a method which computes a whole set of integrals simultaneously where the kernels are transformed into tensorized structure combined with a high-dimensional SVD (Singular Value Decomposition). Before presenting our results, related works are in order. Holst *et al* [6] are prominent specialists for nonlinear PBE by using multi-level FEM (Finite Element Method) solvers. They have treated nonlinear solute-solvent problems on realistic biochemical geometries. Using BEM (Boundary Element Methods) has an advantage over FEM [6, 7, 8, 9] in the linear case because most FEM programs and analysis assume that the right hand side (RHS) is a sufficiently smooth function whereas one has in the solvent problem a RHS which is a sum of nonsmooth Diracs defined only in the sense of distribution leading to extremely fine adaptive FEM-mesh refinements. In addition, the related transmission problem is solved by FEM in the whole 3D space while only the solution on the molecular surface is required. In order to compute the polarization energy, one needs exclusively the electrostatic potential over the surface $\mathbf{\Gamma}$ and not on the whole domain $\mathbf{\Omega}$. On the other hand, most methods based exclusively on BEM which treat the PBE consider only the linear PBE because it is difficult to use fundamental solutions to convert the initial nonlinear PDE (Partial Differential Equation) into integral equations. Concerning the simulation in nanoscale, we have computed [8] the material properties of nanotubes inserted inside a polymer matrix by using a FEM-based approach. An important component of PBE simulation is the geometric information because exact solutions of PBE are only known for very few simple geometries such as single spheres or cylinders. For other more complicated geometries, one must use numerical methods. Implementing a program for generating an SES (Solvent Excluded Surface) [10, 11] from nuclei coordinates is not straightforward because a lot of geometric tasks are involved. It is a long process to start from the nuclei coordinates, passing through mesh processing [12, 11], till obtaining the geometric data using NURBS (Non-Uniform Rational B-Spline) for computations [13, 14, 15, 10]. Beside the patch-based wavelet-BEM [10, 11, 16], alternative methods for boundary integral equations include Fast-Multipole and Panel-Clustering which are mesh-based numerical approximations. A BEM-simulation is reported in [17] where we used a domain decomposition technique to remedy the large condition number of the single layer potential. Every subdomain in the overlapping domain decomposition is constituted of a set of patches. In this paper, we consider the PBE for the interaction of solute and solvent media which are respectively denoted by $\mathbf{\Omega}^{\rm int}$ and $\mathbf{\Omega}^{\rm ext}$. The surface $\mathbf{\Gamma}$ represents the solute-solvent interface [18] which is in our case the molecular surface. The solvent is composed of a continuous dielectric medium containing mobile ions while the solute is located inside the cavity where fixed charges reside. In the sequel, the whole solute-solvent domain is denoted by $\mathbf{\Omega} := \mathbf{\Omega}^{\rm int} \cup \mathbf{\Omega}^{\rm ext}$. This article is based upon BEM which is derived from the integral equations residing on the interface surface $\mathbf{\Gamma}$. We develop a unified approach for all BEM-integrals which are involved for both the nBEM (normal-BEM) and dBEM (derivative-BEM) formulations [19, 20, 21, 22]. The presented approach is valid for integral equations originating from other problems related to partial differential equations but we present it only for molecular models.

2. Wavelet bases for Poisson-Boltzmann

This section will consider the BEM using wavelets on NURBS patches to formulate the Poisson-Boltzmann equation. We consider N fixed atoms $\{\mathbf{x}_k\}_{k=1}^{N}$ which are located inside the solute region $\mathbf{\Omega}^{\rm int}$ while the solvent $\mathbf{\Omega}^{\rm ext}$ is composed of mobile ions as

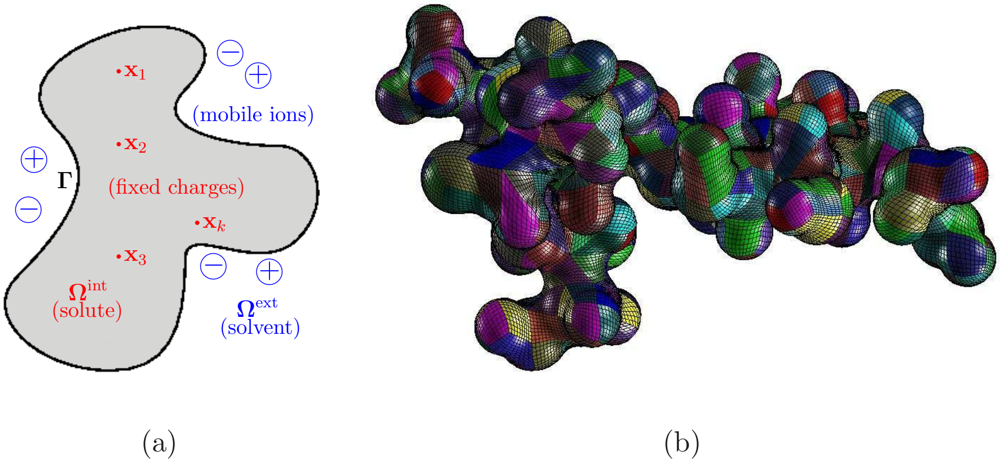


FIGURE 1. (a)Ionic solution (b)Patched molecular manifold deduced from nuclei coordinates.

illustrated in Figure 1(a). We let q_k denote the electric charge of the k-th atom. The treatment of a quantum solvation involves two interacting media which are the solute and the solvent having different permittivities. The principal unknown u is the total electrostatic potential which is the sum of the mobile and fixed electrostatic potentials. In the most general analytical formulation, the nonlinear PBE is expressed by the next partial differential equation

$$-\nabla \cdot \big(\varepsilon(\mathbf{x})\nabla u(\mathbf{x})\big) + \kappa(\mathbf{x})\mathcal{N}\big[u(\mathbf{x})\big] = F(\mathbf{x}) \qquad \forall\, \mathbf{x} \in \mathbf{\Omega} \tag{2.1}$$

The coefficients $\varepsilon(\mathbf{x})$ and $\kappa(\mathbf{x})$ are space-dependent functions which might be discontinuous between $\mathbf{\Omega}^{\text{int}}$ and $\mathbf{\Omega}^{\text{ext}}$ but the solution u is required to be continuous everywhere. The function $\mathcal{N}(\cdot)$ is a real valued univariate nonlinear map. The right hand side $F(\mathbf{x})$ is a sum of Dirac charges centered at the nuclei coordinates $\{\mathbf{x}_k\}_{k=1}^{N}$ inside the solute whereas F vanishes in the solvent. For several counter-ions [23] of number $N_{C.I.}$,

$$\mathcal{N}\big[u(\mathbf{x})\big] = \sum_{i=1}^{N_{C.I.}} n_i Q_i \exp\big[-\beta Q_i u(\mathbf{x})\big] \tag{2.2}$$

where n_i is the density number of counter-ions of type i and Q_i is its charge while β is a parameter involving the temperature. In the case that there are only two counter-ions and $Q_1 = Q_2 = 1/\beta$ while $2n = \beta$, equation (2.2) becomes $\mathcal{N}\big[u(\mathbf{x})\big] = \frac{n}{\beta}\Big(\exp[u(\mathbf{x})] - \exp[-u(\mathbf{x})]\Big) = \sinh\big[u(\mathbf{x})\big]$. By considering only the first term of the Taylor expansion $\sinh(t) = t + t^3/3! + t^5/5! + \cdots$, one deduces the linear PBE. Henceforth, the space dependent factors are assumed to be piecewise constant such as

$$\varepsilon(\mathbf{x}) = \begin{cases} \varepsilon_{\rm int} & \text{for} \quad \mathbf{x} \in \mathbf{\Omega}^{\rm int} \\ \varepsilon_{\rm ext} & \text{for} \quad \mathbf{x} \in \mathbf{\Omega}^{\rm ext}, \end{cases} \qquad \kappa(\mathbf{x}) = \begin{cases} \kappa & \text{for} \quad \mathbf{x} \in \mathbf{\Omega}^{\rm int} \\ 0 & \text{for} \quad \mathbf{x} \in \mathbf{\Omega}^{\rm ext} \end{cases} \tag{2.3}$$

where $\varepsilon_{\rm int}$ and $\varepsilon_{\rm ext}$ are related to the permittivities of the solute and solvent respectively while the scaling factor κ is physically related to the Debye-Hückel parameter. We denote the values of u in the solute and the solvent by $u^{\rm int}$ and $u^{\rm ext}$ respectively. The point of departure for the linearized PBE is governed by the following transmission problem

$$-\varepsilon_{\rm int}\nabla^2 u^{\rm int}(\mathbf{x}) = \sum_{k=1}^{N} q_k \delta(\mathbf{x}-\mathbf{x}_k) \quad \text{for} \quad \mathbf{x} \in \mathbf{\Omega}^{\rm int}, \tag{2.4}$$

$$-\varepsilon_{\rm ext}\nabla^2 u^{\rm ext}(\mathbf{x}) + \kappa u^{\rm ext}(\mathbf{x}) = 0 \quad \text{for} \quad \mathbf{x} \in \mathbf{\Omega}^{\rm ext}, \tag{2.5}$$

$$u^{\rm int}(\mathbf{x}) = u^{\rm ext}(\mathbf{x}) \qquad \text{for} \quad \mathbf{x} \in \mathbf{\Gamma} = \partial\mathbf{\Omega}^{\rm int} \tag{2.6}$$

$$\varepsilon_{\rm int}\frac{\partial}{\partial \mathbf{n}(\mathbf{x})} u^{\rm int}(\mathbf{x}) = \varepsilon_{\rm ext}\frac{\partial}{\partial \mathbf{n}(\mathbf{x})} u^{\rm ext}(\mathbf{x}) \qquad \text{for} \quad \mathbf{x} \in \mathbf{\Gamma} \tag{2.7}$$

where $\delta(\cdot - \mathbf{x}_k)$ is the Dirac distribution centered at the coordinates of the nucleus $\mathbf{x}_k$. We are not solving those equations directly, rather we consider only the pertaining integral equations which are located on the interface surface $\mathbf{\Gamma}$. We consider the PBE on a molecular surface $\mathbf{\Gamma}$ which consists of a set of globally continuous NURBS surfaces that approximate the SES surface.

Since both the wavelet basis function and the mappings $\boldsymbol{\gamma}_p$ are expressed in term of B-spline basis, we shall recall briefly some important properties of a B-spline setting which represents piecewise polynomials. Consider two integers n, k such that $n \geq k \geq 1$. Should the interval $[a, b]$ be the domain of definition of the B-spline, that interval is subdivided by a knot sequence $\boldsymbol{\zeta} = (\zeta_i)_{i=0}^{n+k}$ such that $\zeta_i < \zeta_{i+1}$ for $i = k-1, ..., n-1$ and such that the initial and the final entries of the knot sequence are clamped $\zeta_0 = \cdots = \zeta_{k-1} = a$ and $\zeta_n = \cdots = \zeta_{n+k} = b$. One defines the B-splines [24] basis functions as

$$N_i^{\boldsymbol{\zeta},k}(t) := (\zeta_{i+k} - \zeta_i)[\zeta_i, ..., \zeta_{i+k}](\cdot - t)_+^{k-1} \qquad \text{for } i = 0, ..., n \tag{2.8}$$

where one employs the divided difference $[\zeta_i, \zeta_{i+1}, ..., \zeta_p]f$ in which one uses the truncated power functions $(\cdot - t)_+^k$ given by $(x-t)_+^k := (x-t)^k$ if $x \geq t$, while it is zero otherwise. The integer n controls the number of B-spline functions. Each B-spline basis $N_i^{\boldsymbol{\zeta},k}$ is supported by $[\zeta_i, \zeta_{i+k}] \subset [0, 1]$. The NURBS patch $\boldsymbol{\gamma}_p$ admitting the control points $\mathbf{d}_{i,j} \in \mathbb{R}^3$ and weights $w_{i,j} \in \mathbb{R}^+$ is expressed as

$$\boldsymbol{\gamma}_p(u, v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{n} w_{i,j}\mathbf{d}_{i,j} N_i^{\boldsymbol{\zeta},k}(u) N_j^{\boldsymbol{\zeta},k}(v)}{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j} N_i^{\boldsymbol{\zeta},k}(u) N_j^{\boldsymbol{\zeta},k}(v)} \in \mathbb{R}^3 \qquad \forall\,(u, v) \in \square. \tag{2.9}$$

Each NURBS patch is defined on the unit square $\square := [0,1]^2$. That is, the Connolly [25] surface $\mathbf{\Gamma}$ embedded in $\mathbb{R}^3$ will be decomposed into M patches [11] admitting the following properties:

- We have a covering of the molecular surface by patches $\mathbf{\Gamma} = \bigcup_{p=1}^{M} \mathbf{\Gamma}_p$,
- Each patch $\mathbf{\Gamma}_p$ where $p = 1, 2, \ldots, M$ is the image $\mathbf{\Gamma}_p = \boldsymbol{\gamma}_p(\square)$ such that each $\boldsymbol{\gamma}_p : \square \to \mathbf{\Gamma}_p$ is described by a bivariate NURBS function which is bijective, sufficiently smooth and admitting bounded Jacobians,
- The intersection of two different patches $\mathbf{\Gamma}_p$ and $\mathbf{\Gamma}_q$ is supposed to be either empty, a common curvilinear edge or a common vertex,
- The patch decomposition has a global continuity: for each pair of patches $\mathbf{\Gamma}_p$, $\mathbf{\Gamma}_q$ sharing a curvilinear edge, the parametric representation is subject to a matching condition. That is, a bijective affine mapping $\mathbf{\Xi} : \square \to \square$ exists such that for all $\mathbf{x} = \boldsymbol{\gamma}_p(\mathbf{s})$ on the common curvilinear edge, one has $\boldsymbol{\gamma}_p(\mathbf{s}) = (\boldsymbol{\gamma}_q \circ \mathbf{\Xi})(\mathbf{s})$. In other words, the NURBS functions $\boldsymbol{\gamma}_p$ and $\boldsymbol{\gamma}_q$ agree pointwise at common edges up to some reorientation,
- The manifold $\mathbf{\Gamma}$ is orientable and the normal vector $\mathbf{n}(\mathbf{x})$ is consistently pointing outward for any $\mathbf{x} \in \mathbf{\Gamma}$.

An instance of such a geometric representation for a molecular surface is displayed on Figure 1(b). A comprehensive description of obtaining the above NURBS structure is beyond the scope of this paper. Interested readers are refered to our earlier works [13, 10, 11] for details. The two unknown functions defined on the molecular surface $\mathbf{\Gamma}$ are denoted by

$$u(\mathbf{x}) := u^{\text{int}}(\mathbf{x}) = u^{\text{ext}}(\mathbf{x}) \qquad \text{and} \qquad g(\mathbf{x}) := \frac{\partial u^{\text{int}}(\mathbf{x})}{\partial \mathbf{n}(\mathbf{x})} \quad \text{for} \quad \mathbf{x} \in \mathbf{\Gamma}. \tag{2.10}$$

For two points $\mathbf{x}$, $\mathbf{y}$ belonging to the molecular surface $\mathbf{\Gamma}$, introduce the kernels

$$\mathcal{K}_1(\mathbf{x},\mathbf{y}) := \frac{1}{4\pi}\frac{1}{\|\mathbf{x}-\mathbf{y}\|}, \qquad \mathcal{K}_2(\mathbf{x},\mathbf{y}) := \frac{1}{4\pi}\frac{\partial}{\partial \mathbf{n}(\mathbf{y})}\frac{1}{\|\mathbf{x}-\mathbf{y}\|}, \tag{2.11}$$

$$\mathcal{K}_3(\mathbf{x},\mathbf{y}) := \frac{1}{4\pi}\frac{e^{-\kappa\|\mathbf{x}-\mathbf{y}\|}}{\|\mathbf{x}-\mathbf{y}\|}, \qquad \mathcal{K}_4(\mathbf{x},\mathbf{y}) := \frac{1}{4\pi}\frac{\partial}{\partial \mathbf{n}(\mathbf{y})}\frac{e^{-\kappa\|\mathbf{x}-\mathbf{y}\|}}{\|\mathbf{x}-\mathbf{y}\|}. \tag{2.12}$$

In virtue of the Green's second identity and the fundamental solutions, one has for $\mathbf{x} \in \mathbf{\Gamma}$ two equations related to $\mathbf{\Omega}^{\text{int}}$ and $\mathbf{\Omega}^{\text{ext}}$:

$$\frac{1}{2}u^{\text{int}}(\mathbf{x}) - \int_{\mathbf{\Gamma}} \mathcal{K}_1(\mathbf{x},\mathbf{y})\frac{\partial u^{\text{int}}(\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} d_{\mathbf{\Gamma}}\mathbf{y} + \int_{\mathbf{\Gamma}} \mathcal{K}_2(\mathbf{x},\mathbf{y}) u^{\text{int}}(\mathbf{y}) d_{\mathbf{\Gamma}}\mathbf{y} = \frac{1}{\varepsilon_{\text{int}}}\sum_{k=1}^{N}\frac{q_k}{4\pi\|\mathbf{x}-\mathbf{x}_k\|} \tag{2.13}$$

$$\frac{1}{2}u^{\text{ext}}(\mathbf{x}) + \int_{\mathbf{\Gamma}} \mathcal{K}_3(\mathbf{x},\mathbf{y})\frac{\partial u^{\text{ext}}(\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} d_{\mathbf{\Gamma}}\mathbf{y} - \int_{\mathbf{\Gamma}} \mathcal{K}_4(\mathbf{x},\mathbf{y}) u^{\text{ext}}(\mathbf{y}) d_{\mathbf{\Gamma}}\mathbf{y} = 0. \tag{2.14}$$

Due to the interface conditions (2.6) and (2.7), the above account yields the nBEM as follows.

$$\begin{cases} \frac{1}{2}u(\mathbf{x}) - \int_{\mathbf{\Gamma}} \mathcal{K}_1(\mathbf{x},\mathbf{y})g(\mathbf{y})d_{\mathbf{\Gamma}}\mathbf{y} + \int_{\mathbf{\Gamma}} \mathcal{K}_2(\mathbf{x},\mathbf{y})u(\mathbf{y})d_{\mathbf{\Gamma}}\mathbf{y} = \frac{1}{\varepsilon_{\text{int}}}\sum_{k=1}^{N}\frac{q_k}{4\pi\|\mathbf{x}-\mathbf{x}_k\|} & (2.15)\\ \frac{1}{2}u(\mathbf{x}) + \frac{\varepsilon_{\text{int}}}{\varepsilon_{\text{ext}}}\int_{\mathbf{\Gamma}} \mathcal{K}_3(\mathbf{x},\mathbf{y})g(\mathbf{y})d_{\mathbf{\Gamma}}\mathbf{y} - \int_{\mathbf{\Gamma}} \mathcal{K}_4(\mathbf{x},\mathbf{y})u(\mathbf{y})d_{\mathbf{\Gamma}}\mathbf{y} = 0. & (2.16)\end{cases}$$

A combination of (2.15) and (2.16) yields

$$\frac{1}{2}\left[1+\frac{\varepsilon_{\text{ext}}}{\varepsilon_{\text{int}}}\right] \quad u(\mathbf{x}) - \int_{\mathbf{\Gamma}}[\mathcal{K}_1(\mathbf{x},\mathbf{y}) - \mathcal{K}_3(\mathbf{x},\mathbf{y})]g(\mathbf{y})d_{\mathbf{\Gamma}}\mathbf{y} +$$
$$+ \quad \int_{\mathbf{\Gamma}}[\mathcal{K}_2(\mathbf{x},\mathbf{y}) - \frac{\varepsilon_{\text{ext}}}{\varepsilon_{\text{int}}}\mathcal{K}_4(\mathbf{x},\mathbf{y})]u(\mathbf{y})d_{\mathbf{\Gamma}}\mathbf{y} = \frac{1}{\varepsilon_{\text{int}}}\sum_{k=1}^{N}\frac{q_k}{4\pi\|\mathbf{x}-\mathbf{x}_k\|}.$$

By taking the normal derivative with respect to $\mathbf{n}(\mathbf{x})$ of (2.14) and using $\partial u^{\text{ext}}(\mathbf{x})/\partial\mathbf{n}(\mathbf{x}) = (\varepsilon_{\text{int}}/\varepsilon_{\text{ext}})g(\mathbf{x})$, obtain

$$\frac{1}{2}\frac{\varepsilon_{\text{int}}}{\varepsilon_{\text{ext}}}g(\mathbf{x}) + \frac{\varepsilon_{\text{int}}}{\varepsilon_{\text{ext}}}\int_{\mathbf{\Gamma}}\left[\frac{\partial}{\partial\mathbf{n}(\mathbf{x})}\mathcal{K}_3(\mathbf{x},\mathbf{y})\right]g(\mathbf{y})d_{\mathbf{\Gamma}}\mathbf{y} - \int_{\mathbf{\Gamma}}\left[\frac{\partial}{\partial\mathbf{n}(\mathbf{x})}\mathcal{K}_4(\mathbf{x},\mathbf{y})\right]u(\mathbf{y})d_{\mathbf{\Gamma}}\mathbf{y} = 0.$$

Summing the last equation with $\partial(2.15)/\partial\mathbf{n}(\mathbf{x})$ results in the dBEM:

$$(2.17)\quad \begin{cases} \frac{1}{2}\left[1+\frac{\varepsilon_{\text{ext}}}{\varepsilon_{\text{int}}}\right]u(\mathbf{x}) + \int_{\mathbf{\Gamma}}\widetilde{\mathcal{K}}_1(\mathbf{x},\mathbf{y})u(\mathbf{y})d_{\mathbf{\Gamma}}\mathbf{y} + \int_{\mathbf{\Gamma}}\widetilde{\mathcal{K}}_2(\mathbf{x},\mathbf{y})g(\mathbf{y})d_{\mathbf{\Gamma}}\mathbf{y} = \\ \qquad = \frac{1}{\varepsilon_{\text{int}}}\sum_{k=1}^{N}\frac{q_k}{4\pi\|\mathbf{x}-\mathbf{x}_k\|} \\ \frac{1}{2}\left[1+\frac{\varepsilon_{\text{int}}}{\varepsilon_{\text{ext}}}\right]g(\mathbf{x}) + \int_{\mathbf{\Gamma}}\widetilde{\mathcal{K}}_3(\mathbf{x},\mathbf{y})u(\mathbf{y})d_{\mathbf{\Gamma}}\mathbf{y} + \int_{\mathbf{\Gamma}}\widetilde{\mathcal{K}}_4(\mathbf{x},\mathbf{y})g(\mathbf{y})d_{\mathbf{\Gamma}}\mathbf{y} = \\ \qquad = \frac{1}{\varepsilon_{\text{int}}}\sum_{k=1}^{N}\frac{q_k}{4\pi}\frac{\partial}{\partial\mathbf{n}(\mathbf{x})}\left[\frac{1}{\|\mathbf{x}-\mathbf{x}_k\|}\right]\end{cases}$$

in which one utilizes the next kernels:

$$\begin{aligned}
\widetilde{\mathcal{K}}_1(\mathbf{x},\mathbf{y}) &:= \frac{1}{4\pi}\frac{\partial}{\partial\mathbf{n}(\mathbf{y})}\left[\frac{1}{\|\mathbf{x}-\mathbf{y}\|}\right] - \frac{1}{4\pi}\frac{\varepsilon_{\text{ext}}}{\varepsilon_{\text{int}}}\frac{\partial}{\partial\mathbf{n}(\mathbf{y})}\left[\frac{e^{-\kappa\|\mathbf{x}-\mathbf{y}\|}}{\|\mathbf{x}-\mathbf{y}\|}\right],\\
\widetilde{\mathcal{K}}_2(\mathbf{x},\mathbf{y}) &:= \frac{1}{4\pi}\frac{e^{-\kappa\|\mathbf{x}-\mathbf{y}\|}}{\|\mathbf{x}-\mathbf{y}\|} - \frac{1}{4\pi}\frac{1}{\|\mathbf{x}-\mathbf{y}\|},\\
\widetilde{\mathcal{K}}_3(\mathbf{x},\mathbf{y}) &:= \frac{1}{4\pi}\frac{\partial^2}{\partial\mathbf{n}(\mathbf{x})\partial\mathbf{n}(\mathbf{y})}\left[\frac{1}{\|\mathbf{x}-\mathbf{y}\|}\right] - \frac{1}{4\pi}\frac{\partial^2}{\partial\mathbf{n}(\mathbf{x})\partial\mathbf{n}(\mathbf{y})}\left[\frac{e^{-\kappa\|\mathbf{x}-\mathbf{y}\|}}{\|\mathbf{x}-\mathbf{y}\|}\right],\\
\widetilde{\mathcal{K}}_4(\mathbf{x},\mathbf{y}) &:= \frac{1}{4\pi}\frac{\varepsilon_{\text{int}}}{\varepsilon_{\text{ext}}}\frac{\partial}{\partial\mathbf{n}(\mathbf{x})}\left[\frac{e^{-\kappa\|\mathbf{x}-\mathbf{y}\|}}{\|\mathbf{x}-\mathbf{y}\|}\right] - \frac{1}{4\pi}\frac{\partial}{\partial\mathbf{n}(\mathbf{x})}\left[\frac{1}{\|\mathbf{x}-\mathbf{y}\|}\right].
\end{aligned}$$

The dBEM formulation is numerically more stable than the formulation using nBEM [20, 21]. The Galerkin variational formulation with respect to a finite dimensional space spanned by $(\varphi_\alpha)_{\alpha=1}^n$ uses the approximating functions $u_h(\mathbf{x}) = \sum_{\alpha=1}^n u_\alpha \varphi_\alpha(\mathbf{x})$, $g_h(\mathbf{x}) = \sum_{\alpha=1}^n g_\alpha \varphi_\alpha(\mathbf{x})$, where $\mathbf{U}^T := [u_1, ..., u_n] \in \mathbb{R}^n$ and $\mathbf{G}^T := [g_1, ..., g_n] \in \mathbb{R}^n$ are the BEM-unknowns. The next linear system is eventually obtained

$$\left\{ \begin{bmatrix} \frac{1}{2}[1+\varepsilon_{\text{ext}}/\varepsilon_{\text{int}}]\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{2}[1+\varepsilon_{\text{int}}/\varepsilon_{\text{ext}}]\mathbf{I} \end{bmatrix} + \begin{bmatrix} \widetilde{\mathbf{K}}_1 & \widetilde{\mathbf{K}}_2 \\ \widetilde{\mathbf{K}}_3 & \widetilde{\mathbf{K}}_4 \end{bmatrix} \right\} \begin{bmatrix} \mathbf{U} \\ \mathbf{G} \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{S} \end{bmatrix} \tag{2.18}$$

such that the matrix entries are

$$\mathbf{I}_{(\alpha,\beta)} := \int_{\mathbf{\Gamma}} \varphi_\alpha(\mathbf{x})\varphi_\beta(\mathbf{x}) d_{\mathbf{\Gamma}}\mathbf{x} = \sum_{p=1}^M \int_{\mathbf{\Gamma}_p} \varphi_\alpha(\mathbf{x})\varphi_\beta(\mathbf{x})\, d_{\mathbf{\Gamma}_p}\mathbf{x} \tag{2.19}$$

$$\widetilde{\mathbf{K}}_{i,(\alpha,\beta)} := \sum_{p=1}^M \sum_{q=1}^M \int_{\mathbf{\Gamma}_p} \int_{\mathbf{\Gamma}_q} \widetilde{K}_i(\mathbf{x},\mathbf{y})\varphi_\alpha(\mathbf{x})\varphi_\beta(\mathbf{y}) d_{\mathbf{\Gamma}_p}\mathbf{x}\, d_{\mathbf{\Gamma}_q}\mathbf{y} \tag{2.20}$$

while the term in the right hand side involves

$$R_\alpha := \frac{1}{\varepsilon_{\text{int}}} \sum_{p=1}^M \sum_{k=1}^N \int_{\mathbf{\Gamma}_p} \frac{q_k}{4\pi} \frac{\varphi_\alpha(\mathbf{x})}{\|\mathbf{x}-\mathbf{x}_k\|} d_{\mathbf{\Gamma}_p}\mathbf{x} \tag{2.21}$$

$$S_\alpha := \frac{1}{4\pi\varepsilon_{\text{int}}} \sum_{p=1}^M \sum_{k=1}^N \int_{\mathbf{\Gamma}_p} \frac{\partial}{\partial \mathbf{n}(\mathbf{x})} \left[\frac{q_k \varphi_\alpha(\mathbf{x})}{\|\mathbf{x}-\mathbf{x}_k\|} \right] d_{\mathbf{\Gamma}_p}\mathbf{x}. \tag{2.22}$$

In general, the linear system (2.18) is troublesome because the basis functions $(\varphi_\alpha)_{\alpha=1}^n$ yield dense matrices for the operators $\widetilde{\mathbf{K}}_1$, $\widetilde{\mathbf{K}}_2$, $\widetilde{\mathbf{K}}_3$ and $\widetilde{\mathbf{K}}_4$. In term of memory, if $(\varphi_\alpha)_{\alpha=1}^n$ are standard polynomial basis functions, each matrix $\widetilde{\mathbf{K}}_i$ requires n^2 storage to accommodate all entries. In addition, the determination of a matrix entry of $\widetilde{\mathbf{K}}_i$ calculates an integration in $4D$ where the integrand is highly nonlinear and possibly singular depending on the patches $\mathbf{\Gamma}_p \times \mathbf{\Gamma}_q$. By using tensor product B-spline wavelet basis functions, the matrices $\widetilde{\mathbf{K}}_i$ become quasi-sparse. For the instance of high-order odd LMQ-wavelets [2, 3] as illustrated on Figure 2(a), we depict on Figure 2(b) the quasi-sparse matrix entries for a $2D$ singular operator where very small entries are set to zero. The Polarizable Continuum Model consists in determining the polarization energy $E_{\mathbf{POLARIZATION}} := 0.5 \sum_{k=1}^N q_k\, \phi_{\mathbf{RXN}}(\mathbf{x}_k)$ where $\phi_{\mathbf{RXN}}$ designates the reaction electrostatic potential which is the difference between the electrostatic potentials in gas and in the vacuum:

$$\phi_{\mathbf{RXN}}(\mathbf{x}_k) = \int_{\mathbf{\Gamma}} \left[\frac{\varepsilon_{\text{ext}}}{\varepsilon_{\text{int}}} \big(\mathcal{K}_4(\mathbf{x}_k,\mathbf{y}) - \mathcal{K}_2(\mathbf{x}_k,\mathbf{y}) \big) \right] u(\mathbf{y}) + \Big[\mathcal{K}_1(\mathbf{x}_k,\mathbf{y}) - \mathcal{K}_3(\mathbf{x}_k,\mathbf{y}) \Big] g(\mathbf{y}) d_{\mathbf{\Gamma}}\mathbf{y}.$$

The involved integrals in equation (2.19–2.22) and the reaction electrostatic potential take the form

$$\int_{\mathbf{\Gamma}_p} \int_{\mathbf{\Gamma}_q} \mathcal{K}(\mathbf{x},\mathbf{y})\varphi_\alpha(\mathbf{x})\varphi_\beta(\mathbf{y}) d_{\mathbf{\Gamma}_p}\mathbf{x} d_{\mathbf{\Gamma}_q}\mathbf{y}, \tag{2.23}$$

$$\int_{\mathbf{\Gamma}_p} \mathcal{H}(\mathbf{x})\varphi_\alpha(\mathbf{x})\varphi_\beta(\mathbf{x})d_{\mathbf{\Gamma}_p}\mathbf{x}, \qquad \int_{\mathbf{\Gamma}_p} \mathcal{R}(\mathbf{x})\varphi_\alpha(\mathbf{x})d_{\mathbf{\Gamma}_p}\mathbf{x}. \tag{2.24}$$

With respect to the patches $\mathbf{\Gamma}_p \times \mathbf{\Gamma}_q$, we use tensor product bases

$$\varphi_\alpha(\mathbf{x}) = (\psi_i \otimes \psi_j)(u_1, u_2), \qquad \varphi_\beta(\mathbf{y}) = (\psi_k \otimes \psi_\ell)(v_1, v_2) \tag{2.25}$$

such that for $\mathbf{x} \in \mathbf{\Gamma}_p$ and $\mathbf{y} \in \mathbf{\Gamma}_q$

$$\begin{aligned} \mathbf{x} &= (x_1, x_2, x_3) = \big[\gamma_{p,1}(\mathbf{u}), \gamma_{p,2}(\mathbf{u}), \gamma_{p,3}(\mathbf{u})\big] = \boldsymbol{\gamma}_p(\mathbf{u}) \quad \text{where} \quad \mathbf{u} = (u_1, u_2) \in \square, \\ \mathbf{y} &= (y_1, y_2, y_3) = \big[\gamma_{q,1}(\mathbf{v}), \gamma_{q,2}(\mathbf{v}), \gamma_{q,3}(\mathbf{v})\big] = \boldsymbol{\gamma}_q(\mathbf{v}) \quad \text{where} \quad \mathbf{v} = (v_1, v_2) \in \square. \end{aligned}$$

As a result, the integrals in (2.23) and (2.24) can be replaced by

$$\int_\square \int_\square \mathcal{K}\big[\boldsymbol{\gamma}_p(\mathbf{u}), \boldsymbol{\gamma}_q(\mathbf{v})\big](\psi_i \otimes \psi_j)(u_1, u_2)(\psi_k \otimes \psi_\ell)(v_1, v_2) \\ \left\| \frac{\partial \boldsymbol{\gamma}_p(\mathbf{u})}{\partial u_1} \times \frac{\partial \boldsymbol{\gamma}_p(\mathbf{u})}{\partial u_2} \right\| \left\| \frac{\partial \boldsymbol{\gamma}_q(\mathbf{v})}{\partial v_1} \times \frac{\partial \boldsymbol{\gamma}_q(\mathbf{v})}{\partial v_2} \right\| du_1\, du_2\, dv_1\, dv_2,$$

$$\int_\square \mathcal{H}\big[\boldsymbol{\gamma}_p(\mathbf{u})\big](\psi_i \otimes \psi_j)(u_1, u_2)(\psi_k \otimes \psi_\ell)(u_1, u_2) \left\| \frac{\partial \boldsymbol{\gamma}_p(\mathbf{u})}{\partial u_1} \times \frac{\partial \boldsymbol{\gamma}_p(\mathbf{u})}{\partial u_2} \right\| du_1\, du_2, \tag{2.26}$$

$$\int_\square \mathcal{R}\big[\boldsymbol{\gamma}_p(\mathbf{u})\big](\psi_i \otimes \psi_j)(u_1, u_2) \left\| \frac{\partial \boldsymbol{\gamma}_p(\mathbf{u})}{\partial u_1} \times \frac{\partial \boldsymbol{\gamma}_p(\mathbf{u})}{\partial u_2} \right\| du_1\, du_2. \tag{2.27}$$

We define in $4D$ a function $\mathcal{A}$ which associates to $(t_1, t_2, t_3, t_4) \in [0,1]^4$ the value

$$\mathcal{A}(t_1, t_2, t_3, t_4) := \mathcal{K}\Big[\boldsymbol{\gamma}_p(t_1, t_2), \boldsymbol{\gamma}_q(t_3, t_4)\Big] \left\| \frac{\partial \boldsymbol{\gamma}_p(t_1, t_2)}{\partial t_1} \times \frac{\partial \boldsymbol{\gamma}_p(t_1, t_2)}{\partial t_2} \right\| \\ \left\| \frac{\partial \boldsymbol{\gamma}_q(t_3, t_4)}{\partial t_3} \times \frac{\partial \boldsymbol{\gamma}_q(t_3, t_4)}{\partial t_4} \right\|$$

whereas in $2D$ for $(t_1, t_2) \in [0,1]^2$

$$\mathcal{A}(t_1, t_2) := \mathcal{H}\big[\boldsymbol{\gamma}_p(t_1, t_2)\big] \left\| \frac{\partial \boldsymbol{\gamma}_p(t_1, t_2)}{\partial t_1} \times \frac{\partial \boldsymbol{\gamma}_p(t_1, t_2)}{\partial t_2} \right\|. \tag{2.28}$$

In any dimension D, we compute many integrals for a set of multi-indices $(k_1, ..., k_D)$

$$\mathcal{J}(k_1, ..., k_D) = \int_{\square \times \square} \mathcal{A}(t_1, ..., t_D)\, (\psi_{k_1} \otimes \cdots \otimes \psi_{k_D})(t_1, ..., t_D)\, dt_1 ... dt_D. \tag{2.29}$$

Bivariate functions $\mathcal{A}$ are involved for the computation of the right-hand side, the matrix identity operator and the reaction electrostatic potential whereas 4-variate functions are involved for the operators $\widetilde{\mathbf{K}}_i$. For the $4D$-integrals, we concentrate on the case where the patches $\mathbf{\Gamma}_p$ and $\mathbf{\Gamma}_q$ are disjoint. For the singular case where $\mathbf{\Gamma}_p \cap \mathbf{\Gamma}_q \neq \emptyset$, either some other method can be used, or we avoid division by zero where a kernel involving $1/\|\mathbf{x} - \mathbf{y}\|$ is replaced by $1/\sqrt{\|\mathbf{x} - \mathbf{y}\|^2 + \delta}$ for $\mathbf{x} \in \mathbf{\Gamma}_p$, $\mathbf{y} \in \mathbf{\Gamma}_q$, and an adjustable small parameter $\delta > 0$. A similar approach has been used in [26] for the purpose of quantum mechanics using high-dimensional approximation. As a multi-dimensional domain of integration, we can take a general hypercube $\mathcal{B} = \prod_{i=1}^{D}[a_i, b_i]$ in (2.29) because the multivariate function $\mathcal{A}$ is in practice coupled with some KD-tree subdivision. Next, we consider the construction of wavelet bases on the whole molecular surface. In addition, we survey the properties of the wavelet integrals for the former kernels.

We would like now to survey the construction of the wavelet spaces on the surface $\mathbf{\Gamma}$. Since every two incident patches admit pointwise joints, constructing the wavelet bases on the unit square $\square$ is sufficient to construct basis functions on the whole molecular surface. Each $1D$-wavelet basis will be constructed as a linear combination of B-splines

$$\psi(t) = \sum_{i=0}^{n} q_i N_i^{\boldsymbol{\zeta},k}(t) \in \mathbb{R} \qquad \text{where} \qquad \forall\, t \in [0,1]. \tag{2.30}$$

The integer k in (2.8) controls the polynomial degree $k-1$ of the B-spline which admits an overall smoothness of $\mathcal{C}^{k-2}$ where the case $k = 1$ corresponds to discontinuous piecewise constant functions needed for the Haar wavelet. On level ℓ, we define a knot sequence $\boldsymbol{\zeta}^{\ell} = (\zeta_i^{\ell}) \subset [0,1]$ such that $\zeta_i^{\ell} = 0$ and $\zeta_{n+1+i}^{\ell} = 1$ for $i = 0, ..., k$ and that the remaining $\zeta_i^{\ell} \in]0,1[$. The internal knots on the next level $(\ell+1)$ are obtained by inserting one new knot inside two consecutive knots on the lower level ℓ. Introduce the B-spline linear space on level ℓ:

$$\mathbb{V}_{\ell}^{k}[0,1] := \Big\{ N_i^{\ell} := N_i^{\boldsymbol{\zeta}^{\ell},k} \in \mathbb{L}_2[0,1] : \quad i = 0, ..., n \Big\}. \tag{2.31}$$

By using the piecewise polynomial property of the B-splines and the inclusion $\boldsymbol{\zeta}^{\ell} \subset \boldsymbol{\zeta}^{\ell+1}$, the B-spline bases form a nested sequence of subspaces:

$$\mathbb{V}_{0}^{k}[0,1] \subset \mathbb{V}_{1}^{k}[0,1] \subset \cdots \subset \mathbb{V}_{L}^{k}[0,1] \subset \mathbb{L}_2[0,1]. \tag{2.32}$$

As a consequence, the space $\mathbb{V}_{\ell}^{k}[0,1]$ can be expressed as an orthogonal sum

$$\mathbb{V}_{\ell}^{k}[0,1] = \mathbb{V}_{\ell-1}^{k}[0,1] \oplus \mathbb{W}_{\ell-1}^{k}[0,1] \tag{2.33}$$

with respect to the $\mathbb{L}_2$-scalar product where $\mathbb{W}_{\ell}^{k}[0,1]$ is the wavelet space

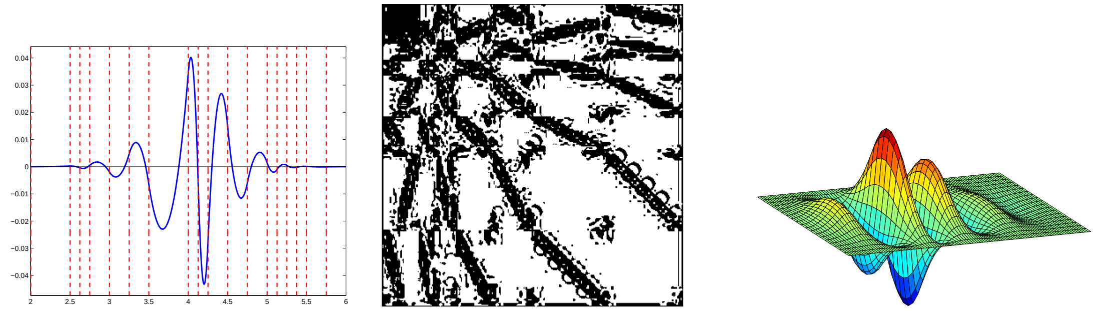
$$\mathbb{W}_{\ell-1}^{k}[0,1] = \text{span}\Big\{ \psi_i^{\ell-1} \in \mathbb{V}_{\ell}^{k}[0,1], \quad \langle \psi_i^{\ell-1}, \phi \rangle_{\mathbb{L}^2[0,1]} = 0, \quad \forall \phi \in \mathbb{V}_{\ell-1}^{k}[0,1] \Big\}. \tag{2.34}$$

By applying the decomposition (2.33) recursively, one obtains on the maximal level L

$$\mathbb{V}_{L}^{k}[0,1] = \mathbb{V}_{0}^{k}[0,1] \oplus \Big(\bigoplus_{\ell=0}^{L-1} \mathbb{W}_{\ell}^{k}[0,1] \Big). \tag{2.35}$$

The $2D$-wavelet spaces (see Figure 2(c)) on the unit square $\square$ is defined for any maximal level L as follows.

$$\mathbb{V}_{L}^{k}(\square) := \mathbb{V}_{L}^{k}[0,1] \otimes \mathbb{V}_{L}^{k}[0,1] = \mathbb{A}^{k}(\square) \oplus \mathbb{B}_{L}^{k}(\square) \oplus \mathbb{C}_{L}^{k}(\square) \oplus \mathbb{D}_{L}^{k}(\square) \tag{2.36}$$



(a) (b) (c)

FIGURE 2. (a)Higher-order wavelet on a nonuniform knot sequence, (b)Quasi-sparse matrix, (c)Bivariate tensor product basis.

such that

$$\mathbb{A}^k(\square) := \mathbb{V}_0^k[0,1] \otimes \mathbb{V}_0^k[0,1], \qquad \mathbb{B}_L^k(\square) := \bigoplus_{\ell=0}^{L-1} \big(\mathbb{W}_\ell^k[0,1] \otimes \mathbb{V}_0^k[0,1]\big),$$

$$\mathbb{C}_L^k(\square) := \bigoplus_{\ell=0}^{L-1} \big(\mathbb{V}_0^k[0,1] \otimes \mathbb{W}_\ell^k[0,1]\big), \qquad \mathbb{D}_L^k(\square) := \bigoplus_{\ell=0}^{L-1}\bigoplus_{m=0}^{L-1} \big(\mathbb{W}_\ell^k[0,1] \otimes \mathbb{W}_m^k[0,1]\big).$$

We want now to survey the determination of the wavelet bases ψ_i^ℓ of $\mathbb{W}_\ell^k[0,1]$. The construction of the higher-order LMQ-wavelets [2, 3] requires the case on the whole infinite real line $\mathbb{R}$ on which one has knot entries on each integer as $\zeta_i := i$. The cardinal B-spline is given by

$$N_i^{\mathrm{CARDINAL}}(x) = \int_0^1 N_{i-1}^{\mathrm{CARDINAL}}(x-t)dt \tag{2.37}$$

which verifies the two scale relation

$$N_i^{\mathrm{CARDINAL}}(x) = \sum_{j=0}^{i} 2^{-i+1}\binom{j}{i} N_i^{\mathrm{CARDINAL}}(2x-j). \tag{2.38}$$

For the polynomial degree k, the corresponding complementary space is spanned by the shifts of the wavelet function

$$\psi_{\mathrm{CARDINAL}}(x) := \frac{1}{2^{k-1}} \sum_{j=0}^{2m-2} (-1)^j N_{2k}^{\mathrm{CARDINAL}}(j+1) \frac{d^k}{dx_k} N_{2k}^{\mathrm{CARDINAL}}(2x-j). \tag{2.39}$$

Representing the derivatives $d^k N_{2k}^{\mathrm{CARDINAL}}(2x-j)/dx_k$ in term of B-splines can be used to express the function ψ_{CARDINAL} in term of control points. The cardinal wavelets are orthogonal to B-splines on the infinite real line $\mathbb{R}$ having integers as knot sequence. The above cardinal splines serve as construction of internal wavelets on the interval $[0,1]$ by scaling and shifting. We describe only the odd wavelets where the polynomial degree k is odd because the even case is defined almost similarly with the exception of additional central wavelets. We define on level ℓ the clamped knot sequence $\zeta_0^\ell = \cdots = \zeta_{k-1}^\ell = 0$, and $\zeta_{n_\ell}^\ell = \cdots = \zeta_{n_\ell+k}^\ell = 1$ where $n_\ell := 2^\ell - 1$. We sometimes drop the dependence on ℓ to simplify the notations. The $1D$-wavelets use the following internal knot sequence $\boldsymbol{\zeta}^{(\ell)}$

$$\zeta_j^\ell := \Big(\Big\lfloor \frac{k}{2} \Big\rfloor + j\Big) 2^{-\ell} \in]0,1[, \qquad \text{for} \qquad j = k, ..., n_\ell. \tag{2.40}$$

One defines the *internal* wavelet functions by means of scaled and dilated transformations

$$\psi_j^\ell(t) := \psi_{\text{CARDINAL}}\big(2^\ell t - (j-k+1)\big). \tag{2.41}$$

Additionally, we need *boundary* wavelet functions which are of the form

$$\psi_j^\ell(t) = \sum_{p=0}^{k+\lfloor \frac{k}{2}\rfloor -2} q_{p,\ell} N_p^{\ell+1}(t) + \sum_{p=k+\lfloor \frac{k}{2}\rfloor -1}^{k+\lfloor \frac{k}{2}\rfloor +2\ell} q_{p,\ell} N_p^{\ell+1}(t) \tag{2.42}$$

where the coefficients $q_{p,\ell}$ for the last summation are taken from the coefficients of the cardinal wavelet ψ_{CARDINAL}. The remaining unknown coefficients $q_{p,\ell}$ on the first summation are obtained by solving

$$\sum_{p=0}^{k+\lfloor \frac{k}{2}\rfloor -2} \big\langle N_p^{\ell+1}, N_m^\ell \big\rangle_{\mathbb{L}_2[0,1]} q_{p,\ell} = - \sum_{p=k+\lfloor \frac{k}{2}\rfloor -1}^{k+\lfloor \frac{k}{2}\rfloor +2\ell} q_{p,\ell} \big\langle N_p^{\ell+1}(t), N_m^\ell \big\rangle_{\mathbb{L}_2[0,1]}, \tag{2.43}$$

for $m = 0, ..., k + \lfloor \frac{k}{2} \rfloor - 2$. That ensures the $\mathbb{L}_2$-orthogonality

$$\langle \psi_j^\ell, N_m^\ell \rangle_{\mathbb{L}_2[0,1]} = 0, \qquad \forall N_m^\ell \in \mathbb{V}_\ell^k[0,1]. \tag{2.44}$$

The previous construction creates $2^{\ell-1}$ wavelet basis functions. To complete the construction, the remaining ones are deduced by symmetry such as $\psi_j^\ell(t) := \psi_{2^\ell-j-1}^\ell(1-t)$ for $t \in [0,1]$, $j = 2^{\ell-1}, ..., 2^\ell - 1$. The construction of the Haar wavelets and piecewise linear wavelets is very similar. The former corresponds to piecewise constant polynomials while the latter to piecewise polynomials of degree $k = 1$. Since every polynomial of degree $(k-1)$ can be expressed in term of N_i^ℓ, we obtain from the orthogonality (2.44) the property of vanishing moments:

$$\int_0^1 (t - \widetilde{t})^p \psi_j^\ell(t) dt = 0, \qquad \text{for} \qquad 0 \le p \le k-1 \tag{2.45}$$

for any fixed $\widetilde{t}$. It is an important property because it yields quasi-sparse structure of the matrices $\widetilde{\mathbf{K}}_i$ from (2.18). We examine the value of $\mathcal{A}(\mathbf{u}, \mathbf{v})$ from (2.29) on a pair of patches $\mathbf{\Gamma}_p \times \mathbf{\Gamma}_q$. By fixing some $(\widetilde{\mathbf{u}}, \widetilde{\mathbf{v}}) = [(\tilde{u}_1, \tilde{u}_2), (\tilde{v}_1, \tilde{v}_2)] \in \Box \times \Box$ and by considering any $(\mathbf{u}, \mathbf{v}) = [(u_1, u_2), (v_1, v_2)] \in \Box \times \Box$, the Taylor expansion yields

$$\mathcal{A}(\mathbf{u},\mathbf{v}) \;=\; \sum_{|\boldsymbol{\alpha}|+|\boldsymbol{\beta}|\leq k-1} \frac{1}{\boldsymbol{\alpha}!}\frac{1}{\boldsymbol{\beta}!}\frac{\partial^{|\boldsymbol{\alpha}|}}{\partial \mathbf{u}^{\boldsymbol{\alpha}}}\frac{\partial^{|\boldsymbol{\beta}|}}{\partial \mathbf{v}^{\boldsymbol{\beta}}}\mathcal{A}(\widetilde{\mathbf{u}},\widetilde{\mathbf{v}})(\mathbf{u}-\widetilde{\mathbf{u}})^{\boldsymbol{\alpha}}(\mathbf{v}-\widetilde{\mathbf{v}})^{\boldsymbol{\beta}} + \tag{2.46}$$

$$\sum_{|\boldsymbol{\alpha}|+|\boldsymbol{\beta}|=k} R^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{u},\mathbf{v})(\mathbf{u}-\widetilde{\mathbf{u}})^{\boldsymbol{\alpha}}(\mathbf{v}-\widetilde{\mathbf{v}})^{\boldsymbol{\beta}}. \tag{2.47}$$

For the first summation, by multiplication with a tensor product wavelet basis $\boldsymbol{\psi}^p \otimes \boldsymbol{\psi}^q$=$(\psi_1^p \otimes \psi_2^p)$ $\otimes$$(\psi_1^q \otimes \psi_2^q)$ and by taking the integration over $\Box \times \Box$, one obtains for $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ and $\boldsymbol{\beta} = (\beta_1, \beta_2)$ where $|\boldsymbol{\alpha}| = |\alpha_1|+|\alpha_2| \leq k-1$ and $|\boldsymbol{\beta}| = |\beta_1|+|\beta_2| \leq k-1$

$$\frac{1}{\boldsymbol{\alpha}!}\frac{1}{\boldsymbol{\beta}!}\frac{\partial^{|\boldsymbol{\alpha}|}}{\partial \mathbf{u}^{\boldsymbol{\alpha}}}\frac{\partial^{|\boldsymbol{\beta}|}}{\partial \mathbf{v}^{\boldsymbol{\beta}}}\mathcal{A}(\widetilde{\mathbf{u}},\widetilde{\mathbf{v}}) \int_0^1 (u_1-\widetilde{u}_1)^{\alpha_1}\psi_1^p(u_1)du_1 \cdots \int_0^1 (v_2-\widetilde{v}_2)^{\beta_2}\psi_2^q(v_2)dv_2 \tag{2.48}$$

which is zero due to the property of the vanishing moment (2.45). As for the second summation, introduce $\delta_r := \max_{i=1,2}\big\{|u_i - \tilde{u}_i|, \quad \text{where} \quad (u_1,u_2) \in \text{Support}(\boldsymbol{\psi}^r)\big\}$ for $r = p, q$. The summation is estimated by $\sum_{|\boldsymbol{\alpha}|+|\boldsymbol{\beta}|=k}\big|R_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{u},\mathbf{v})\big| \max_{\mathbf{u}\in\Box}\big|\boldsymbol{\psi}^p(\mathbf{u})\big|$ $\max_{\mathbf{v}\in\Box}\big|\boldsymbol{\psi}^q(\mathbf{v})\big|\ \delta_p^{|\boldsymbol{\alpha}|}\delta_q^{|\boldsymbol{\beta}|}$. By defining $\delta := \max\{\delta_p, \delta_q\} < 1$, one obtains

$$\delta^k \sum_{|\boldsymbol{\alpha}|+|\boldsymbol{\beta}|=k} \max_{(\mathbf{u},\mathbf{v})\in\Box\times\Box}\big|R^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{u},\mathbf{v})\big| \max_{\mathbf{u}\in\Box}\big|\boldsymbol{\psi}^p(\mathbf{u})\big| \max_{\mathbf{v}\in\Box}\big|\boldsymbol{\psi}^q(\mathbf{v})\big|. \tag{2.49}$$

Supposing that the Jacobians of the mappings $\boldsymbol{\gamma}_p$ and $\boldsymbol{\gamma}_q$ are bounded, one obtains for $\mathbf{x} = \boldsymbol{\gamma}_p(\mathbf{u})$ and $\mathbf{y} = \boldsymbol{\gamma}_q(\mathbf{v})$. On account of the following generalized Calderon-Zygmund inequality for the involved kernels in the case $|\boldsymbol{\alpha}| + |\boldsymbol{\beta}| = k$:

$$\big|R^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{u},\mathbf{v})\big| \leq \Big|\frac{\partial^{|\boldsymbol{\alpha}|}}{\partial \mathbf{x}^{\boldsymbol{\alpha}}}\frac{\partial^{|\boldsymbol{\beta}|}}{\partial \mathbf{y}^{\boldsymbol{\beta}}}\mathcal{K}(\mathbf{x},\mathbf{y})\Big| \leq C\frac{1}{\|\mathbf{x}-\mathbf{y}\|^{k+1}}, \quad \forall\,(\mathbf{x},\mathbf{y}) \in \mathbf{\Gamma}_p \times \mathbf{\Gamma}_q \tag{2.50}$$

which is small if the images $\boldsymbol{\gamma}_p\big[\text{Support}(\boldsymbol{\psi}^p)\big]$ and $\boldsymbol{\gamma}_q\big[\text{Support}(\boldsymbol{\psi}^q)\big]$ are sufficiently distant from one another. Thus, the speed of the decay toward zero depends on the factor δ^k using the vanishing moments exponent k and the distances between the basis supports. Hence, the constructed wavelet basis has the advantage that it renders the operators $\widetilde{\mathbf{K}}_i$ quasi-sparse.

3. Wavelet integral tensorization

In the sequel, we describe the computation of the wavelet integrals by means of higher order singular value decompositions for tensors of arbitrary orders. The objective is to simultaneously evaluate a lot of integrals corresponding to wavelet bases given by a set of multi-indices. We recall only some tensor techniques which are needed for our wavelet application as a lengthy comprehensive treatment is found in [27]. For $I_1, ..., I_D \in \mathbb{N}$, a tensor of order D is denoted as $X \in \mathbb{R}^{I_1\times\cdots\times I_D}$ having entries $X[i_1, ..., i_D] \in \mathbb{R}$ where $i_p \in I_p$ for $p = 1, ..., D$. In particular, a tensor of second order is a matrix. An n-mode product of a D-ordered tensor X and a matrix $A \in \mathbb{R}^{I \times I_n}$ is defined as the tensor $X \times_n A \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times I \times I_{n+1} \times \cdots \times I_N}$ such that

$$(X \times_n A)[j_1, ..., j_{n-1}, i, j_{n+1}, ..., j_D] = \sum_{j_n=1}^{I_n} X[j_1, j_2, ..., j_n, ..., j_D] A[i, j_n] \tag{3.51}$$

for $i = 1, ..., I$. The n-mode product fulfills the associativity property $X \times_m A \times_n B = (X \times_m A) \times_n B = (X \times_n B) \times_m A$. The Kronecker product of two matrices A, B is defined as

$$A \otimes B := \begin{bmatrix} a_{1,1}B & \dots & a_{1,n}B \\ \vdots & & \vdots \\ a_{m,1}B & \dots & a_{m,n}B \end{bmatrix}. \tag{3.52}$$

The matricization of a tensor $X \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ is the creation of a matrix $\underline{X}_{(\mathcal{R} \times \mathcal{C})}$ by organizing the multi-indices into two sets of rows and columns $\mathcal{R} = \{r_1, ..., r_L\}$, $\mathcal{C} = \{c_1, ..., c_M\}$ forming a partition of $\mathcal{N} = \{1, ..., D\}$

$$(\underline{X}_{(\mathcal{R} \times \mathcal{C})})[j, k] := X[i_1, i_2, ..., i_N] \tag{3.53}$$

such that the entries $[j, k]$ has the lexicographic ordering w.r.t. $\mathcal{R}$ and $\mathcal{C}$:

$$j = 1 + \sum_{\ell=1}^{L} \big[(i_{r_\ell} - 1) \prod_{\ell'=1}^{\ell-1} I_{r_{\ell'}}\big], \qquad k = 1 + \sum_{m=1}^{M} \big[(i_{c_m} - 1) \prod_{m'=1}^{m-1} I_{c_{m'}}\big]. \tag{3.54}$$

The matrix $\underline{X}_{(\mathcal{R} \times \mathcal{C})}$ admits as numbers of rows and columns: $I_{\mathcal{R}} := \prod_{n \in \mathcal{R}} I_n$, $I_{\mathcal{C}} := \prod_{n \in \mathcal{C}} I_n$. The n-mode matricization is the special case when one has the row subset is a singleton $\mathcal{R} = \{n\}$ and $\mathcal{C} = \{1, ..., n-1, n+1, ..., D\}$ such that $\underline{X}_{(n)} := \underline{X}_{(\mathcal{R} \times \mathcal{C})}$. One has the equivalence $X = Y \times_n A$ if and only if $\underline{X}_{(n)} = A \underline{Y}_{(n)}$. The tensor X has a Tucker decomposition $(G, A^{(1)}, ..., A^{(D)})$ where $G \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ if

$$X = G \times_1 A^{(1)} \times_2 A^{(2)} \times_3 \cdots \times_D A^{(D)} \tag{3.55}$$

which is equivalent to

$$\underline{X}_{(\mathcal{R} \times \mathcal{C})} = \Big[A^{(r_L)} \otimes \cdots \otimes A^{(r_1)}\Big] (\underline{Y}_{(\mathcal{R} \times \mathcal{C})}) \Big[A^{(c_M)} \otimes \cdots \otimes A^{(c_1)}\Big]. \tag{3.56}$$

In term of n-mode, this means for each $n = 1, ..., D$

$$\underline{X}_n = A^{(1)} \underline{G}_n \Big[A^{(D)} \otimes \cdots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \cdots \otimes A^{(1)}\Big] \tag{3.57}$$

In term of tensor coefficients, the Tucker decomposition is described as

$$X[i_1, ..., i_D] = \sum_{j_1=1}^{I_1} \cdots \sum_{j_D=1}^{I_D} G[j_1, ..., j_D] A^{(1)}[i_1, j_1] \cdots A^{(D)}[i_D, j_D] \tag{3.58}$$

which is not unique in general. In the case that the tensor is such that $G \in \mathbb{R}^{R\times\cdots\times R}$ is the identity tensor, one obtains the Kruskal decomposition where all the matrices $A^{(n)}$ admit R columns. Thus, the Kruskal decomposition is such that

$$X = [\![A^{(1)}, \cdots, A^{(D)}]\!] \tag{3.59}$$

where one defines the tensor

$$\left[\!\!\left[A^{(1)}, \cdots, A^{(D)}\right]\!\!\right][i_1, ..., i_D] := \sum_{r=1}^{R} A^{(1)}[i_1, r] \cdots A^{(D)}[i_D, r]. \tag{3.60}$$

The Kruskal decomposition (3.59) is the generalization to tensors of arbitrary order of the usual SVD for matrices. In fact, $X = U\sigma V^T$ amounts to $X[i_1, i_2] = \sum_{j_1} \sum_{j_2} \sigma[j_1, j_2]$ $U[i_1, j_1]V[i_2, j_2]$. Since the matrix σ is diagonal, multiplying $\sigma[j, j]$ into one of the matrices U, V provides the decomposition (3.60) for the case $D = 2$. If R is minimal, then it is the rank of the tensor X. It is possible for a tensor of a general order $D \geq 3$ that the rank is larger that $\min\{I_1, ..., I_D\}$. Suppose we have a D-ordered tensor $X \in \mathbb{R}^{I_1\times\cdots\times I_D}$ such that

$$\mathcal{A}_h(t_1, ..., t_D) = \sum_{i_1=1}^{I_1} \cdots \sum_{i_D=1}^{I_D} X[i_1, ..., i_D] \left(\bigotimes_{p=1}^{D} N_{i_p}^{\boldsymbol{\zeta}^p} \right) (t_1, ..., t_D) \tag{3.61}$$

where $\bigotimes_{p=1}^{D} N_{i_p}^{\boldsymbol{\zeta}^p}$ are tensor product B-spline bases functions. If the tensor X admits a Kruskal decomposition (3.59), then the function (3.61) becomes

$$\mathcal{A}_h = \sum_{i_1=1}^{I_1} \cdots \sum_{i_D=1}^{I_D} \left[\!\!\left[A^{(1)}, \cdots, A^{(D)}\right]\!\!\right] \left(\bigotimes_{p=1}^{D} N_{i_p}^{\boldsymbol{\zeta}^p} \right). \tag{3.62}$$

We want to evaluate the integrals using the basis functions

$$\psi_{k_1}^{(1)} \otimes \cdots \otimes \psi_{k_D}^{(D)} \quad \text{where} \quad (k_1, ..., k_D) \in \mathcal{F} \tag{3.63}$$

in which $\mathcal{F}$ is a set of multi-indices that is not necessarily a tensor product set. In practice, the set $\mathcal{F}$ is determined by a collection of wavelet bases involved in the Beylkin quasi-sparse structure but we do not consider that issue in this paper. Consider the domain of integration $\mathcal{B}_D := \prod_{i=1}^{D} [a_i, b_i]$ of $\psi_{k_1}^{(1)} \otimes \cdots \otimes \psi_{k_D}^{(D)}$. The integrals from the former sections can be expressed as

$$\begin{aligned} \left\langle \mathcal{A}_h, \bigotimes_{p=1}^{D} \psi_{k_p}^{(p)} \right\rangle_{\mathcal{B}_D} = \sum_{r=1}^{R} & \left[\sum_{i_1=1}^{I_1} A^{(1)}[i_1, r] \int_{a_1}^{b_1} N_{i_1}^{\boldsymbol{\zeta}^1}(t_1) \psi_{k_1}^{(1)}(t_1) dt_1 \right] \cdots \\ & \cdots \left[\sum_{i_D=1}^{I_D} A^{(D)}[i_D, r] \int_{a_D}^{b_D} N_{i_D}^{\boldsymbol{\zeta}^p}(t_D) \psi_{k_D}^{(D)}(t_D) dt_D \right]. \end{aligned}$$

Thus, instead of computing high-dimensional integrals, they are reduced to $1D$-integrals which are easy to compute. The $1D$-integrals can be computed exactly once for all and they are only accessed subsequently by some table look-up.

```
Algorithm: tensorized integral computation
INPUTS: $\mathcal{A}_h = \sum_{i_1=1}^{I_1} \cdots \sum_{i_D=1}^{I_D} X[i_1, ..., i_D] \left( \bigotimes_{p=1}^{D} N_{i_p}^{\boldsymbol{\zeta}^p} \right)$
        Multi-index set $\mathcal{F}$ such that $\mathbf{k} = (k_1, ..., k_D) \in \mathcal{F}$
OUTPUTS: Multi-dimensional integrals $\left\langle \mathcal{A}_h, \bigotimes_{p=1}^{D} \psi_{k_p}^{(p)} \right\rangle_{\mathcal{B}_D}$
 1 : Compute the SVD as $X = [\![A^{(1)}, ..., A^{(D)}]\!]$
 2 : Assemble $\mathcal{F}_p := \{k_p : \exists \mathbf{k} \in \mathcal{F} \text{ where } \mathbf{k} = (..., k_p, ...)\}$ for each $p = 1, .., D$
 3 : for $p = 1, ..., D$:
 4 :     Compute the $1D$-integrals $\int_{a_p}^{b_p} N_i^{\boldsymbol{\zeta}^p}(t) \psi_k^{(p)}(t) dt$ for $i = 1, ..., I_p$ and $k \in \mathcal{F}_p$
 5 :     Assemble the matrices $\mathcal{V}_p[i, k] = \int_{a_p}^{b_p} N_i^{\boldsymbol{\zeta}^p}(t) \psi_k^{(p)}(t) dt$
 6 : end
 7 : for $p = 1, ..., D$:
 8 :     Consider the table look-up $\mathcal{V}_p[i, k]$
 9 :     Compute $\mathcal{U}_p[r, k] = \sum_{i=1}^{I_p} A^{(p)}[i, r] \mathcal{V}_p[i, k]$ for $r = 1, ..., R$ and $k \in \mathcal{F}_p$
10 : end
11 : Release the matrices $\mathcal{V}$
12 : Deduce $\left\langle \mathcal{A}_h, \bigotimes_{p=1}^{D} \psi_{k_p}^{(p)} \right\rangle_{\mathcal{B}_D} = \sum_{r=1}^{R} \prod_{p=1}^{D} \mathcal{U}_p[r, k_p]$ for $\mathbf{k} = (k_1, ..., k_D) \in \mathcal{F}$.
```

In fact, for the set of basis $\mathcal{F}$, we define

$$\mathcal{U}_1(r, k_1) := \sum_{i_1=1}^{I_1} A^{(1)}[i_1, r] \mathcal{V}_1(i_1, k_1), ..., \mathcal{U}_D(r, k_D) := \sum_{i_D=1}^{I_D} A^{(D)}[i_D, r] \mathcal{V}_D(i_D, k_D) \tag{3.64}$$

in which we use the expressions

$$\mathcal{V}_1(i_1, k_1) := \int_{a_1}^{b_1} N_{i_1}^{\boldsymbol{\zeta}^1}(t_1) \psi_{k_1}^{(1)}(t_1) dt_1, ..., \mathcal{V}_D(i_D, k_D) := \int_{a_D}^{b_D} N_{i_D}^{\boldsymbol{\zeta}^D}(t_D) \psi_{k_D}^{(D)}(t_D) dt_D. \tag{3.65}$$

Therefore, we eventually obtain

$$\left\langle \mathcal{A}_h, \bigotimes_{p=1}^{D} \psi_{k_p}^{(p)} \right\rangle_{\mathcal{B}_D} = \sum_{r=1}^{R} \prod_{p=1}^{D} \mathcal{U}_p(r, k_p). \tag{3.66}$$

Based on the formerly presented method, we summarize the simultaneous computations of the integrals corresponding to the set $\mathcal{F}$ in the algorithm on page 15. The reason for decoupling the two loops in lines 3–6 and lines 7–10 is that the first one is independent of a single computation. That is, if that algorithm is applied repeatedly, the first loop needs to be executed only once.

4. Numerical results

This section is devoted for the presentation of the practical results pertaining to the former approach. Before presenting our results, we briefly describe the computer implementation. The formerly proposed method has been implemented in C functions together with C++ classes. All linear operations have been performed with the help of the linear libraries BLAS and LAPACK. In particular, the computations of the singular value decomposition are accomplished with the LAPACK-routine `DGESDD`. The quantum models with NURBS have been created by our previous implementation described in [10, 11, 13]. It accepts a list of nuclei coordinates together with the corresponding Van-der-Waals radii. In addition, some user-defined probe radius is provided in order to obtain globally smooth cavity surfaces. The radii of the probe atoms range from 1.0 Angstrom to 2.5 Angstroms for all the involved quantum models. The visualization has been implemented with the help of OpenGL. The computations have been executed on a computer possessing a 4.1 GHz processor and 32 GB RAM. We employ different sorts of quantum models including water clusters which are in fact obtained from a former MD simulation. When the MD iteration attains its equilibrium state where the total energy (sum of the kinetic energy and the potential energy according to the Hamiltonian equation becomes stable, a water cluster is obtained by extracting the water molecules which are contained in some given large sphere whose radius controls the final size of the water cluster. Therefore, the hydrogen and oxygen atoms contained in that large sphere constitute the components of the water clusters. In the numerical results presented below, we report only on $2D$-integrals using Haar and piecewise linear wavelets but the formerly presented algorithm is valid for more general cases.

As a first test, we want to describe the expense of the different stages which are involved in the method of SVD. Those stages consist of: (1)preprocessing step, (2)assembly of the tensors, (3)the SVD processing. That last step is again subdivided into the computation of the SVD, the multiplication with the side matrices which provides the final integral values. The results of the computational intensity are gather in Table 1 for the case of the Haar wavelets while they are in Table 2 for the case of the piecewise linear wavelets. The total runtimes are also displayed in the same tables where we use a water cluster and a DNA section as quantum models. The preprocessing step consists first in evaluating the 3D images by the NURBS mappings from some uniformly spaced grid. That is useful because evaluating the NURBS for many points at once is much less expensive than evaluating them individually. In addition to the 3D images embedded on the patches, we evaluate also the outward normal vectors and their corresponding norms. We apply that to all patches once for all for subsequent table look-up. In our experience, the acceleration factor which is gained in that simultaneous evaluation is

LEV	Prepro	Tensor	S.V.D.			Total
ELS	cessing		Assembly	$1D$-product	$2D$-product	
Water cluster (1109 patches, 237 nuclei):						
1	0.05 sec	296.31 sec	5.81 sec	0.61 sec	0.46 sec	303.24 sec
2	0.13 sec	301.04 sec	14.25 sec	1.53 sec	0.87 sec	317.82 sec
3	0.37 sec	306.18 sec	36.58 sec	5.99 sec	2.36 sec	351.48 sec
4	1.22 sec	344.33 sec	135.42 sec	22.55 sec	8.13 sec	511.65 sec
DNA (2119 patches, 116 nuclei):						
1	0.09 sec	279.33 sec	5.46 sec	0.62 sec	0.47 sec	285.97 sec
2	0.24 sec	280.97 sec	13.60 sec	1.54 sec	0.90 sec	297.25 sec
3	0.70 sec	288.61 sec	34.79 sec	5.27 sec	2.25 sec	331.62 sec
4	2.32 sec	321.76 sec	124.28 sec	20.70 sec	7.10 sec	476.16 sec

Table 1. Haar wavelets.

in the range of 50-250 depending on the number of control points and the smoothness of the NURBS patches. As exhibited in Table 1, the preprocessing step requires only very short computing time. The stages which require long duration is the computation of the tensors and the assembly of the SVD decomposition. The former seems to grow only very lightly as the multiscale levels increase but the latter grows substantially with the levels. That holds for both the Haar wavelets and the piecewise linear ones. The intensity of the products with the SVD decomposition is not negligible but they are not intensive when compared to the tensor processing.

LEV	Prepro	Tensor	S.V.D.			Total
ELS	cessing		Assembly	$1D$-product	$2D$-product	
Water cluster (1109 patches, 237 nuclei):						
1	0.07 sec	295.20 sec	7.18 sec	1.45 sec	0.95 sec	304.85 sec
2	0.50 sec	312.42 sec	48.06 sec	9.70 sec	2.38 sec	373.06 sec
3	3.59 sec	437.97 sec	463.48 sec	72.31 sec	11.41 sec	988.76 sec
4	3.64 sec	515.43 sec	517.22 sec	92.96 sec	18.05 sec	1147.30 sec
DNA (2119 patches, 116 nuclei):						
1	0.12 sec	277.02 sec	7.25 sec	1.38 sec	0.71 sec	286.48 sec
2	0.96 sec	290.36 sec	45.19 sec	9.80 sec	2.38 sec	348.69 sec
3	7.00 sec	406.46 sec	425.30 sec	68.89 sec	8.16 sec	915.81 sec
4	6.95 sec	474.03 sec	480.13 sec	87.22 sec	13.27 sec	1061.60 sec

Table 2. Piecewise linear wavelets.

We want now to examine the efficiency of the proposed method. Alongside the SVD integration method, we use also the Genz-Malik method [28] which has been implemented in the cubature program [29] for comparison purpose. The cubature tool is in fact a very good integrator on general hypercubes. We have tested it for many analytical functions and it yields very accurate results while using moderately few evaluations of the integrand function. The problem here is that the integrand functions which are involved in the former sections are in general very expensive to evaluate. That is caused by several factors: (1)we have NURBS transformations from the unit square to the patches embedded in the space, (2)we have to evaluate not only the images of the NURBS patches but also the normal vectors, (3)the evaluation of the Jacobian and its square root for the Gram-determinant takes very long when applied a lot of times, (4)when the NURBS patches are smooth and admit many control points then thousands of individual evaluations are very expensive.

LEV	Haar wavelets			Linear wavelets		
ELS	SVD	Cubature	Ratio	SVD	Cubature	Ratio
Propane (75 patches, 11 nuclei):						
1	0.98 sec	0.89 sec	0.9081	1.19 sec	4.02 sec	3.3781
2	1.07 sec	7.56 sec	7.0654	1.19 sec	50.25 sec	42.2268
3	1.13 sec	40.10 sec	35.4867	2.64 sec	196.84 sec	74.5606
4	1.96 sec	166.02 sec	84.7040	3.47 sec	skipped	—
5	4.07 sec	skipped	—	37.06 sec	skipped	—
Water cluster (386 patches, 63 nuclei):						
1	28.22 sec	34.25 sec	1.2136	29.20 sec	112.85 sec	3.8647
2	29.74 sec	232.73 sec	7.8254	34.77 sec	1366.14 sec	39.2907
3	38.87 sec	1261.18 sec	32.4461	93.03 sec	5300.59 sec	56.9772
4	54.91 sec	4107.78 sec	74.8093	104.55 sec	skipped	—
5	111.88 sec	skipped	—	1076.07 sec	skipped	—

Table 3. Runtimes for small molecules.

We would like first to describe small molecules admitting patches of number less than 500 and nuclei of number less than 100. The objective is the comparison of the runtimes for the SVD method on the one hand and the direct numerical quadrature on the other. The results of the tests are displayed in Table 3. For that case, we have as quantum models a propane and a water cluster. The former is constituted of 75 patches and 11 nuclei while the latter admits 386 patches and 63 nuclei. The comparison consists of multiscale levels between 1 and 5 in which we utilize both Haar wavelets and piecewise linear wavelets. In the same table, we display also the ratio between the runtimes needed for the SVD method and the cubature method. We do not know the exact values of the integrals but both methods converge to the same values. Since the results provided by the two independent methods agree very well, we consider as exact values the ones which are given by both of them. We abort the integral computations as soon as the two methods yield comparable accuracies when compared to the exact values. In lower levels, all computations can still be performed for both methods. But for higher levels, not all computations have been completed for the cubature method because it takes so long durations that we could not complete them in some reasonable computing time. The cubature results in that case are skipped as identified in some entries of the table where there is no ratio information either. In particular, for level 5, all computations for the cubature methods are skipped for both the Haar wavelet and the piecewise linear one.

LEV	Haar wavelets			Linear wavelets		
ELS	SVD	Cubature	Ratio	SVD	Cubature	Ratio
Water cluster (1109 patches, 237 nuclei):						
1	303.24 sec	476.64 sec	1.5718	304.85 sec	1249.23 sec	4.0978
2	317.82 sec	3396.28 sec	10.6861	373.06 sec	12795.47 sec	34.2986
3	351.48 sec	12312.63 sec	35.0308	988.76 sec	skipped	—
4	511.65 sec	skipped	—	1147.30 sec	skipped	—
DNA (2119 patches, 116 nuclei):						
1	285.97 sec	436.80 sec	1.5274	286.48 sec	1157.13 sec	4.0391
2	297.25 sec	2961.96 sec	9.9645	348.69 sec	11019.07 sec	31.6013
3	331.62 sec	10786.34 sec	32.5262	915.81 sec	skipped	—
4	476.16 sec	skipped	—	1061.60 sec	skipped	—

Table 4. Runtimes for larger molecules.

In most cases, the SVD methods yield much better performance as compared to the direct quadrature in the case of the Haar wavelets. An exception is the first level of the propane molecule where the cubature taking 0.89 seconds is somewhat quicker than the SVD which lasts 0.98 seconds. But this lower level is not really relevant as both methods are still very fast. The performance of the SVD method becomes more tangible as the multiscale levels become larger as shown by the size of the runtime ratio. In contrast to the Haar wavelets, the out-performance is already visible in the case of piecewise linear wavelets even in very low multiscale levels. Because the size of the ratio starting from level 3 in the case of Haar wavelets and starting from level 2 in the case of piecewise linear wavelets amounts to larger than 30, we assume that this method can reduce the cost of computations from matter of hours to matter of minutes. We perform the same tests but for larger molecules in Table 4 where we consider a water cluster constituting of 1109 NURBS patches and 237 nuclei. We consider there also a section of a DNA structure consisting 2119 patches and 116 nuclei. We use only multiscale levels 1–4 because the involved sizes are large to compute. It turns out that for all the levels involved, the SVD method gives much better results than the direct quadrature rules.

Conclusion

We proposed a method for computing the integrals pertaining to the Poisson-Boltzmann formulation when wavelets bases are used for the discretization. Our method is based on higher-order singular value decomposition of tensors of arbitrary dimension. The approach requires that the molecular surface be subdivided into several smooth four-sided patches which are in our case in NURBS form. When applied to real molecular models, our numerical experience shows that the SVD method achieves an efficient acceleration gain.

References

[1] Cohen, A., Daubechies, I. and Feauveau, J. (1992) Biorthogonal bases of compactly supported wavelets. Comm. Pure Appl. Math., 45(5), 485–560.

[2] Lyche, T. and Mørken, K. (1992) Spline-wavelets of minimal support. Numerical Methods in Approximation Theory, Birkhäuser, Basel, 177–194.

[3] Lyche, T., Mørken, K. and Quak, E. (2001) Theory and algorithms for non-uniform spline wavelets, Multivariate approximation and applications. Cambridge: Cambridge University Press, 152–187.

[4] Beylkin, G. (1992) On the representation of operators in bases of compactly supported wavelets. SIAM J. Numer. Anal., 29(6), 1716–1740.

[5] Kleemann, B., Rathsfeld, A. and Schneider, R. (1996) Multiscale methods for boundary integral equations and their application to boundary value problems in scattering theory and geodesy. Notes Numer. Fluid Mech., 54, 1–28.

[6] Baker, N., Sept, D., Holst, M. and McCammon, J. (2001) The adaptive multilevel finite element solution of the Poisson-Boltzmann equation on massively parallel computers. IBM J. Res. Dev., 45, 427–438.

[7] Randrianarivony, M. (2004) Anisotropic finite elements for the Stokes problem: a-posteriori error estimator and adaptive mesh. J. Comput. Appl. Math., 169(2), 255–275.

[8] Diedrich, C., Dijkstra, D., Hamaekers, J., Henniger, B. and Randrianarivony, M. (2015) A finite element study on the effect of curvature on the reinforcement of matrices by randomly distributed and curved nanotubes. J. Comput. Theor. Nanosci., 12(9), 2108–2116.

[9] Apel, T. and Randrianarivony, M. (2003) Stability of the discretizations of the Stokes problem on anisotropic meshes. Math. Comput. Simulat., 61(3-6), 437–447.

[10] Harbrecht, H. and Randrianarivony, M. (2009) Wavelet BEM on molecular surfaces: parametrization and implementation. Computing, 86, 1–22.

[11] Harbrecht, H. and Randrianarivony, M. (2011) Wavelet BEM on molecular surfaces: Solvent Excluded Surfaces. Computing, 92(4), 335–364.

[12] Randrianarivony, M. (2008) Harmonic variation of edge size in meshing CAD geometries from IGES format. Lect. Notes Comput. Sci., 5102, 56–65.

[13] Randrianarivony, M. and Brunnett, G. (2008) Preparation of CAD and molecular surfaces for meshfree solvers. Lect. Notes Comput. Sci. Eng., 65, 231–245.

[14] Randrianarivony, M. (2009) On global continuity of Coons mappings in patching CAD surfaces. Comput.-Aided Design, 41(11), 782–791.

[15] Randrianarivony, M. (2011) On transfinite interpolations with respect to convex domains. Comput. Aided Geom. Des., 28(2), 135–149.

[16] Weijo, V., Randrianarivony, M., Harbrecht, H. and Frediani, L. (2010) Wavelet formulation of the Polarizable Continuum Model. J. Comput. Chem., 31(7), 1469–1477.

[17] Randrianarivony, M. (2016) Domain decomposition for wavelet single layer on geometries with patches. Applied Mathematics, 7(15), 1798–1823.

[18] Stamm, B., Cances, E., Lipparini, F. and Maday, Y. (2016) A new discretization for the polarizable continuum model within the domain decomposition paradigm. J. Chem. Phys., 144(054101).

[19] Bardhan, J. (2009) Numerical solution of boundary-integral equations for molecular electrostatics. J. Comput. Phys., 130(094102).

[20] Liang, J. and Subramaniam, S. (1997) Computation of molecular electrostatics with boundary element methods. Biophysical Journal, 73, 1830–1841.

[21] Juffer, A., Botta, E., Keulen, B., Ploeg, A. and Berendsen, H. (1991) The electric potential of a macromolecule in a solvent: a fundamental approach. Journal of Computational Physics, 97, 144–171.

[22] Jadhao, V., Solis, F. and Cruz, M. (2013) Free-energy functionals of the electrostatic potential for Poisson-Boltzmann theory. Phys. Rev. E, 88(022305).

[23] Teso, A., Filho, E. and Neto, A. (1997) Solution of the Poisson-Boltzmann equation for a system with four ionic species. Journal of Mathematical Biology, 35, 814–824.

[24] DeBoor, C. and Fix, G. (1973) Spline approximation by quasi-interpolants. J. Approx. Theory, 8, 19–45.

[25] Bajaj, C., Xu, G. and Zhang, Q. (2009) A fast variational method for the construction of adaptive resolution C^2 smooth molecular surfaces. Comput. Methods Appl. Mech. Eng., 198, 1684–1690.

[26] Ammar, A. and Chinesta, F. (2008) Circumventing curse of dimensionality in the solution of highly multidimensional models encountered in quantum mechanics using meshfree finite sums decompositions. Lect. Notes in Comput. Sci. Eng., 65, 1–17.

[27] Kolda, T. and Bader B. (2007) Tensor decomposition and applications. SANDIA National Laboratory Report, 6702, 1–71.

[28] Genz, A. and Malik, A. (1980) An adaptive algorithm for numeric integration over an N-dimensional rectangular region. J. Comput. Appl. Math., 6(4), 295–302.

[29] Johnson, S. (2010), `http://ab-initio.mit.edu/wiki/index.php/Cubature`