

# mFES: A Robust Molecular Finite Element Solver for Electrostatic Energy Computations

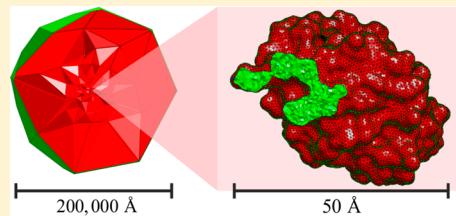
I. Sakalli,<sup>†</sup> J. Schöberl,<sup>‡</sup> and E. W. Knapp\*,<sup>†</sup>

<sup>†</sup>Freie Universität Berlin, Institute of Chemistry and Biochemistry, Fabeckstr. 36a, Berlin 14195, Germany

<sup>‡</sup>Technische Universität Wien, Institute for Analysis and Scientific Computing, Wiedner Hauptstraße 8–10, Vienna 1040, Austria

## S Supporting Information

**ABSTRACT:** We present a robust method for the calculation of electrostatic potentials of large molecular systems using tetrahedral finite elements (FE). Compared to the finite difference (FD) method using a regular simple cubic grid to solve the Poisson equation, the FE method can reach high accuracy and efficiency using an adaptive grid. Here, the grid points can be adjusted and are placed directly on the molecular surfaces to faithfully model surfaces and volumes. The grid point density decreases rapidly toward the asymptotic boundary to reach very large distances with just a few more grid points. A broad set of tools are applied to make the grid more regular and thus provide a more stable linear equation system, while reducing the number of grid points without compromising accuracy. The latter reduces the number of unknowns significantly and yields shorter solver execution times. The accuracy is further enhanced by using second order polynomials as shape functions. Generating the adaptive grid for a molecular system is expensive, but it pays off, if the same molecular geometry is used several times as is the case for  $pK_A$  and redox potential computations of many charge variable groups in proteins. Application of the mFES method is also advantageous, if the molecular system is too large to reach sufficient accuracy when computing the electrostatic potential with conventional FD methods. The program mFES is free of charge and available at <http://agknapp.chemie.fu-berlin.de/mfes>.



## INTRODUCTION

Computation of electrostatic properties of molecular systems is an important method for computational chemistry and dates back to pioneering work of Warwicker and Watson.<sup>1</sup> Evaluating the electrostatic potential and energy of a molecular system, one can analyze, for instance, the interaction of residues in proteins, the energetics of ligand binding and the role of binding pockets, the influence of conformational changes on protein function and also the role of pH dependent interactions in proteins by calculating  $pK_A$  values.<sup>2</sup> In principle, three main methods can be used to compute such properties.

**Finite Difference.** The linearized Poisson–Boltzmann Equation (*lPBE*) is commonly solved with the Finite Difference method [APBS,<sup>3</sup> Delphi,<sup>4</sup> MEAD,<sup>5</sup> UHBD,<sup>6</sup> CPB<sup>7</sup>]. The FD method uses a simple cubic grid and is therefore very robust and of moderate complexity. With the FD method, it is straightforward to define a molecular system with respect to surface and volume albeit the precision is moderate unless the grid is very fine. However, solving the *lPBE*, there are powerful techniques such as focusing to adaptively refine the local accuracy for a molecular system.<sup>8,9</sup> The number of linear equations corresponding to the *lPBE* is equal to the number of grid points [i.e., degrees of freedom (dof)] that grows with  $n^3$ , where  $na$  is the linear extension of the molecular system and  $a$  the lattice constant. An improvement of the FD method was recently implemented by Boschitsch et al.<sup>7</sup> and Mirzadeh et al.<sup>7,10</sup> They refined the cubic grid near the molecular surface by using an adaptive meshing procedure resulting in a high resolution molecular surface and a coarse representation of

solute and solvent volume. With this strategy, they were able to render the FD method more efficient and to avoid focusing.

**Boundary Elements.** Boundary Element (BE) methods employ Green's theorem to solve the *lPBE* by evaluating boundary integral equations on the surface of the molecule.<sup>11–16</sup> Thereby the molecular surface is represented by grid points discretizing the integral equations to solve them numerically. The atomic partial charges of the molecule induce charges on the molecular surface. No discretization of the volume is necessary and therefore no artificial self-interaction energy terms arise.<sup>17</sup> Because these methods rely on a representation of the two-dimensional molecular surface only, the number of dof diminishes considerably compared to the FE and in particular to the FD method.<sup>18</sup> However, the main problem of the BE method is the evaluation of singular boundary integrals with high accuracy.<sup>19,20</sup> The problem with weak-, strong- and hypersingular integrals can be solved with some effort.<sup>19,21</sup> Hence, stable and precise BE methods that solve the PBE may be possible. Additionally, with the BE method computation of forces are possible.<sup>16</sup>

**Finite Elements.** Solving the PBE with the FE method is challenging, since expertise in the field of scientific engineering, mathematical modeling, and computational chemistry has to be combined. One of the first implementations of the FE method for molecules, called PBF<sup>22</sup> (Poisson Boltzmann Finite element method), was developed by Friesner et al.<sup>23,24</sup> and distributed by

Received: June 11, 2014

Published: October 9, 2014



the Schrödinger molecular modeling company where PBF is part of Jaguar software package.<sup>22</sup> These authors use the Galerkin method<sup>24</sup> to solve the PBE and compared molecular electrostatic solvation energies against the FD method of Delphi whose solver uses a Gauss–Seidel iteration with successive over-relaxation (SOR).<sup>4,5,25</sup> This FE method performs well for molecules and small proteins. A limit of 1000 atoms is set because of possible molecular surface discontinuities.<sup>23</sup>

Another FE approach is the Adaptive Finite Element Method (AFEM)<sup>26</sup> that solves the PB equation with the inexact Newton method combined with a multilevel solver.<sup>26–29</sup> Here, the molecular surface is constructed by an iso-contour surface (also called blobby surface<sup>30,31</sup>) based on the sum of atom-centered Gaussian functions. This procedure yields a smooth and robust molecular surface that avoids singular behavior of the PB equation. However, such molecular surfaces may leave small cavities between atoms, which are too small to host solvent molecules, but are filled with the continuum dielectric of the solvent and thus increase electrostatic solvation artificially.<sup>28,32</sup> To generate the tetrahedral volume and solve the PB equation the program suite FETK<sup>33</sup> is used by AFEM<sup>26</sup> involving GAMer<sup>33</sup> and TetGen.<sup>34</sup> Good agreement with the analytical solvable Born ion model<sup>35</sup> was obtained. With a moderate number of refinement steps converged electrostatic solvation energies were obtained for insulin (PDB id 1RWE)<sup>36</sup> involving 1578 atoms.<sup>26</sup> This FE approach is available on the APBS web page,<sup>37</sup> where it is, however, recommended to use the FD method.

An alternative approach was used in FOSLS (first-order system least-squares) by Olson et al.<sup>38</sup> They are using a weighted least-squares FE method, which is also implemented in FETK<sup>33</sup> that uses GAMer<sup>33</sup> to generate molecular models. They obtain converged results for the Born ion model and small molecules using different refinement methods but are facing problems with large proteins. They adaptively refine their molecular model to converge, and the results are very sensitive to details of the protein surface.

Other FE approaches use a mortar FE method to describe molecular systems for solving the *l*PBE<sup>39,40</sup> or even the nonlinear PBE.<sup>29,41</sup> In this method, the volume of the considered system is partitioned in suitable subdomains where the discretization is performed independently. The subdomains are subsequently glued together (with mortar). As a consequence, common boundary faces of volume elements from neighbor subdomains coincide only partially such that the set of grid points defining such boundary faces differ.

We use an adaptive tetrahedral grid to generate the molecular FE model, which work practically without user intervention. They involve the advancing front method from NETGEN<sup>42</sup> and quality improvement steps providing quality information. Different algorithms are available to solve the *l*PBE (see methods section). Based on comparisons of results from protein electrostatics computations, we believe that our FE method is robust and very precise to be competitive with presently available FD methods and superior to available FE methods. Our FE program mFES is freely available at <http://agknapp.chemie.fu-berlin.de/mfes>.

To demonstrate the accuracy of mFES (molecular Finite Element Solver), we compare our results with two widely used finite difference solvers, MEAD<sup>5</sup> (Macroscopic Electrostatics with Atomic Detail) and APBS<sup>3</sup> (Adaptive Poisson–Boltzmann Solver) and two finite element solvers, AFEM<sup>26</sup> (Adaptive Finite Element Method), and PBF<sup>23,24</sup> (Poisson Boltzmann

Finite element method). First, we show a proof of principle by comparing our method and FD solver to a known analytical result, the Born ion model including also ionic strength. Then, we compute electrostatic solvation energies based on electrostatics for bovine pancreatic trypsin inhibitor (bpti) (PDB id 1PIT,<sup>43</sup> involving 899 atoms with hydrogen atoms), barnase (PDB id 1A2P,<sup>44</sup> involving 1724 atoms with hydrogen atoms), lysozyme (PDB id 2LZT,<sup>45</sup> involving 1979 atoms with hydrogen atoms), and cytochrome *c* oxidase (CcO) (PDB id 1MS6,<sup>46</sup> involving 17868 atoms with hydrogen atoms) to show the performance of different methods using proteins with increasing number of atoms. We also provide maps of electrostatic potential and calculate the electrostatic solvation energy for the adenovirus (PDB id 4CWU, involving about 193 000 atoms with hydrogen atoms).<sup>47</sup>

In the present application, we consider also moderate ion concentrations, where the electrostatic energy can still be evaluated approximately by solving the *l*PBE. The definition of the molecular surface is crucial for electrostatic potential calculations. Discretization of a molecule onto an equidistant simple cubic grid is challenging, since for grid boxes at the interface between molecule and solvent, it must be decided whether the grid box volume belongs to the molecule or solvent. A proper decision should depend on the degree of convexity or concavity of the local molecular surface patch. For the FD program APBS, we found empirically that a high point density on the atomic van der Waals (vdW) surfaces is suitable to model local concave surface areas while for convex surface areas low atomic vdW point densities are more suitable to model the molecular surface most faithfully. This aspect is not considered appropriately by most approaches that characterize molecular surfaces by discretization methods.

We first cover the molecular surface by triangles whose shapes are regularized for a given resolution. Second, we fill the protein volume by tetrahedrons at a given resolution, which use the triangles at the molecular surface as faces. We define the asymptotic boundary using a sphere with large radius where the molecule is located in its center and the sphere surface is meshed by few large regular triangles. Third, we fill also the space between the protein and asymptotic sphere surface by tetrahedrons starting from the triangles of the protein surface, enlarging the edge lengths of the tetrahedrons regularly by a grading parameter toward the triangles at the asymptotic sphere surface. The resulting tetrahedral mesh of the model is optimized and simplified keeping the triangles at the protein surface invariant. Finally, the grid points of the tetrahedral mesh are used to discretize the *l*PBE and solve the resulting medium sized linear equation system with a direct method.

## METHODS

**General Overview.** To determine electrostatic energies of a charge distribution  $\rho(\vec{r})$  in an inhomogeneous dielectric medium, we calculate electrostatic potential  $\Phi(\vec{r})$  by solving the *l*PBE

$$\vec{\nabla}[\epsilon(\vec{r})\vec{\nabla}\Phi(\vec{r})] = \kappa^2(\vec{r})\Phi(\vec{r}) - 4\pi\rho(\vec{r}) \quad (1)$$

where  $\epsilon(\vec{r})$  is the spatially dependent dielectric constant and  $\kappa^2(\vec{r})$  the ionic strength. In the following, we will consider the Poisson equation where  $\kappa^2 = 0$ . For molecules, and, in particular, proteins, the charge distribution is defined by  $N_q$  point charges  $q_i$  localized at the atom positions  $\vec{r}_i$

$$\rho(\vec{r}) = \sum_{i=1}^{N_q} q_i \delta(\vec{r} - \vec{r}_i) \quad (2)$$

The electrostatic solvation energy difference  $\Delta G^{\text{vac} \rightarrow \text{solv}}$  of a point charge distribution in two different dielectric environments (vac and solv) is given by

$$\Delta G^{\text{vac} \rightarrow \text{solv}} = G^{\text{solv}} - G^{\text{vac}} = \frac{1}{2} \sum_{i=1}^{N_q} q_i (\Phi_i^{\text{solv}} - \Phi_i^{\text{vac}}) \quad (3)$$

If vac refers to a molecule in vacuum ( $\epsilon_{\text{vac}} = 1$  outside and  $\epsilon_{\text{mol}}$  inside the molecule) and solv to the same molecule in a solvent represented by a dielectric continuum model ( $\epsilon_{\text{solv}}$  outside and  $\epsilon_{\text{mol}} = \epsilon_{\text{protein}}$  inside the molecule), the electrostatic solvation energy difference, eq 3, is the electrostatic contribution to the solvation energy of the considered molecule. For large and flexible molecules such as proteins, we use inside the protein a dielectric constant larger than unity, typically  $\epsilon_{\text{protein}} = 4^{48,49}$  to account for small variations in protein conformation, which are not taken into account explicitly. As a consequence, the reference dielectric constant used for the desolvated protein is  $\epsilon_{\text{homo}} = 4$  everywhere. This setup is used to compute electrostatic solvation energies of proteins in the frame of continuum electrostatics and is used for instance in applications such as p*K*<sub>A</sub> value calculations.<sup>50,51</sup> According to Schulz and Warshel,<sup>49</sup> the dielectric constant inside a protein accounts for "factors that are not considered explicitly". Naturally, if everything could be considered explicitly,  $\epsilon_{\text{protein}} = 1$ .

In part A, we use a weak form of the Poisson equation and show how to describe the electrostatics of a molecular model using a matrix formulation. In part B, the test functions used to expand the electrostatic potential are defined. In part C, we introduce the linear equation solver. Part D describes how an initial molecular surface is generated using a modified version of LSMS,<sup>52</sup> which is a level-set method implementation to calculate the molecular surface. In part E, the molecular surface mesh is generated and refined with NETGEN<sup>42</sup> based on the initial surface. In part F, it is described how NETGEN generates and refines the volume mesh in and outside of a considered molecule. Finally, the obtained volume mesh is optimized with NETGEN as described in part G.

**A. Formulating and Solving the Poisson Equation. Weak Form of the Poisson Equation.** In the weak form, the Poisson equation is solved by minimizing the residual

$$r^e(\vec{r}) = \vec{\nabla}[\epsilon(\vec{r})\vec{\nabla}\Phi(\vec{r})] + 4\pi\rho(\vec{r}) \quad (4)$$

which is performed in the whole volume  $\Omega$  using the integral expression

$$\int_{\Omega} \omega(\vec{r})\vec{\nabla}[\epsilon(\vec{r})\vec{\nabla}\Phi(\vec{r})]d\vec{r} = -4\pi \int_{\Omega} \omega(\vec{r})\rho(\vec{r})d\vec{r} \quad (5)$$

where appropriate test functions  $\omega(\vec{r})$  are used. The integrand on the left side of eq 5 can be transformed using the identity

$$\begin{aligned} & \vec{\nabla}[\omega(\vec{r})\epsilon(\vec{r})\vec{\nabla}\Phi(\vec{r})] \\ &= \epsilon(\vec{r})\vec{\nabla}[\omega(\vec{r})]\vec{\nabla}[\Phi(\vec{r})] + \omega(\vec{r})\vec{\nabla}[\epsilon(\vec{r})\vec{\nabla}\Phi(\vec{r})] \end{aligned} \quad (6)$$

With the Gauss integral equation applied on the left side of eq 6, we obtain

$$\begin{aligned} & \int_{\Omega} \vec{\nabla}[\omega(\vec{r})\epsilon(\vec{r})\vec{\nabla}\Phi(\vec{r})]d\vec{r} \\ &= \int_{\Gamma} \epsilon(\vec{r})\omega(\vec{r})\vec{n}(\vec{r})[\vec{\nabla}\Phi(\vec{r})]d\sigma \end{aligned} \quad (7)$$

where  $\vec{n}(\vec{r})$  is the surface normal vector pointing toward the exterior of the volume  $\Omega$  and  $\Gamma$  is the boundary surface of the volume  $\Omega$ . Using relation 6 and 7, we can rephrase the weak form of the Poisson eq 5 yielding

$$\begin{aligned} & \int_{\Omega} \epsilon(\vec{r})\vec{\nabla}[\omega(\vec{r})]\vec{\nabla}[\Phi(\vec{r})]d\vec{r} \\ &= 4\pi \int_{\Omega} \rho(\vec{r})\omega(\vec{r})d\vec{r} + \int_{\Gamma} \epsilon(\vec{r})\omega(\vec{r})\vec{n}(\vec{r})[\vec{\nabla}\Phi(\vec{r})]d\sigma \end{aligned} \quad (8)$$

which avoids a derivative of  $\epsilon(\vec{r})$ . The whole volume  $\Omega$ , where the Poisson equation needs to be solved, is covered by a sum of small local domains  $\Omega_k$ :  $\Omega = \sum_k \Omega_k$  consisting of tetrahedrons. As test functions  $\omega_j(\vec{r})$ , we use Lagrange interpolation polynomials, which are localized at individual grid points  $\vec{r}_j$  to expand the electrostatic potential

$$\Phi(\vec{r}) = \sum_j \Phi_j \omega_j(\vec{r}) \quad (9)$$

The test functions  $\omega_j(\vec{r})$  have the following properties. They are nonvanishing at their reference grid point  $\vec{r}_j$  and the tetrahedrons  $\Omega_k^{(j)}$  that possess  $\vec{r}_j$  as corner point, but vanish at the other corner points of these tetrahedrons and outside of these tetrahedrons.

Specifying the test function  $\omega(\vec{r})$  in eq 8 to refer to grid point  $i$ ,  $\omega(\vec{r}) \equiv \omega_i(\vec{r})$ , and inserting the expansion (9) in eq 8 we obtain

$$\begin{aligned} \sum_j \sum_k \int_{\Omega_k^{(i,j)}} \epsilon(\vec{r})[\vec{\nabla}\omega_i(\vec{r})\vec{\nabla}\omega_j(\vec{r})]d\vec{r} \Phi_j &= 4\pi \sum_k \int_{\Omega_k^{(i)}} \rho(\vec{r})\omega_i(\vec{r})d\vec{r} \\ &+ \sum_j \sum_k \int_{\Gamma_k^{(i,j)}} \epsilon(\vec{r})\omega_i(\vec{r})[\vec{n}(\vec{r})\vec{\nabla}\omega_j(\vec{r})]d\sigma \Phi_j \end{aligned} \quad (10)$$

where the sums over  $k$  refer to the tetrahedrons  $\Omega_k^{(i,j)}$  that have the common corner points  $\vec{r}_i$  and  $\vec{r}_j$  (or only  $\vec{r}_i$  in case of  $\Omega_k^{(i)}$ ). We can write eq 10 in matrix form

$$(\mathbf{A} + \mathbf{C})\vec{\Phi} = \vec{b} \quad (11)$$

after introducing

$$\begin{aligned} \mathbf{A}_{ij} &= \sum_k \int_{\Omega_k^{(i,j)}} \epsilon(\vec{r})[\vec{\nabla}\omega_i(\vec{r})\vec{\nabla}\omega_j(\vec{r})]d\vec{r}, \\ (\vec{b})_i &= 4\pi \sum_k \int_{\Omega_k^{(i)}} \rho(\vec{r})\omega_i(\vec{r})d\vec{r}, \\ \mathbf{C}_{ij} &= \sum_k \int_{\Gamma_k^{(i,j)}} \epsilon(\vec{r})\omega_i(\vec{r})[\vec{n}(\vec{r})\vec{\nabla}\omega_j(\vec{r})]d\sigma \end{aligned} \quad (12)$$

The matrices  $\mathbf{A}$  and  $\mathbf{C}$  are sparse, since matrix elements are nonzero only if  $\vec{r}_i$  and  $\vec{r}_j$  are equal or are neighbor grid points (being corners of the same tetrahedron). The corresponding integrals involve only contributions from tetrahedrons possessing both grid points  $\vec{r}_i$  and  $\vec{r}_j$  as corner points.

The surface of the protein is described by triangles. These triangles are used as one side of a pair of tetrahedrons, one is localized inside, the second outside of the protein volume. In this way, each tetrahedron covering the space in and outside of a protein cannot simultaneously belong in part to the inside and to the outside of the protein. Hence, the dielectric constant

inside the volume of a tetrahedron is constant but will have different values for the tetrahedrons in and outside of the protein. The values of the dielectric constant of the matrix elements  $A_{ij}$  or  $C_{ij}$  refer to grid points and not to tetrahedrons. If the neighbor grid points  $\vec{r}_i$  and  $\vec{r}_j$ , which may also be equal ( $\vec{r}_i = \vec{r}_j$ ), are both in or outside of the protein the matrix elements have a dielectric constant adopting the value in or outside of the protein. If one of the grid points,  $\vec{r}_i$  and  $\vec{r}_j$  or both are on the protein surface, the effective dielectric constant associated with the matrix element  $A_{ij}$  or  $C_{ij}$  interpolates between the values of the dielectric constants inside the tetrahedrons that possess these grid points as corner points. Since the dielectric constant at a given grid point is the weighted average value of the neighbor tetrahedrons, the matrix  $A$  is symmetric in contrast to more conventional approaches where the matrix  $A$  needs to be symmetrized.

The matrix element  $C_{ij}$  involves surface integrals. These surface integrals may have only nonvanishing contributions for triangular surfaces of tetrahedrons  $\Omega_k^{(ij)}$ , where  $\vec{r}_i$  and  $\vec{r}_j$  are corners of the same triangle. Since each triangle, which contributes to the value of  $C_{ij}$ , belongs to two different tetrahedrons (except for tetrahedrons at the asymptotic boundary), the triangle contributes twice to the integral  $C_{ij}$ . The contributions of the two triangles cancel each other precisely, since the normal surface vectors  $\vec{n}(\vec{r})$  of these triangles have opposite directions. If the electrostatic potential  $\Phi$  vanishes at the asymptotic boundary surface, we can neglect the surface integrals also for triangles at the boundary. Hence, the linear equation [eq 11] simplifies to

$$\mathbf{A}\vec{\Phi} = \vec{b} \quad (13)$$

#### *Eliminating the Grid Artifact Occurring for Point Charges.*

The charge parameters  $\rho_i$  entering eqs 10 and 12 are the result of linear interpolations of the atomic point charges representing the charge distribution in the considered molecular system. The resulting fractional charges on different grid points from the same atom start to interact. For high resolution grids these artificial interactions are much larger than the interactions between different atomic point charges and are thus hiding the latter. This so-called grid artifact generally occurs if atomic point charges are redistributed on grid points and is typically also appearing with FD approaches to solve the *l*PBE. There are, however, techniques that can remove the singular behavior from point charges in the *l*PBE equation explicitly using the reaction field.<sup>53</sup>

In applications, one is usually interested in differences of electrostatic energies as for instance the electrostatic solvation energy, which is the difference between a molecule in solvent ( $\epsilon_{\text{solv}}$ ) and in vacuum ( $\epsilon_{\text{vac}}$ ); see eq 3. In the general case, the dielectric constant in a molecule  $\epsilon_{\text{mol}}$  may differ from unity as explained above. Hence, in terms of the dielectric constants the electrostatic solvation energy, eq 3, can formally be rewritten as

$$\Delta G^{\text{vac} \rightarrow \text{solv}} = G^{\text{vac}}(\epsilon_{\text{vac}}, \epsilon_{\text{mol}}) - G^{\text{solv}}(\epsilon_{\text{solv}}, \epsilon_{\text{mol}})$$

Solving the *l*PBE numerically for both terms in the above expression using precisely the same grid, the grid artifact is the same in both terms and cancels in the difference. This procedure is commonly used also in FD approaches. In case the reaction field is used<sup>53</sup> as implemented in AFEM<sup>26</sup> and PBF,<sup>23,24</sup> the grid artifact does not occur and a single evaluation of the *l*PBE is sufficient.

**B. Expanding the Electrostatic Potential with Test Functions.** The local test functions  $\omega_j(\vec{r})$  used for the expansion,

eq 9 of the electrostatic potential  $\Phi(\vec{r})$  are sums of Lagrange polynomials

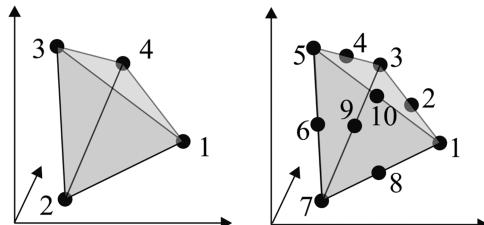
$$\omega_j(\vec{r}) = \sum_{k(j)} p_{k(j)}(\vec{r}) \quad (14)$$

one for each tetrahedron, which has one corner at the grid point  $\vec{r}_j$ . The coefficients of the Lagrange polynomials  $p_{k(j)}(\vec{r})$  are determined by the geometry of the tetrahedrons  $\Omega_k^{(j)}$ , which possess the grid point  $\vec{r}_j$  as a corner point. In first order ( $\sigma = 1$ ), we use linear polynomials with four known coefficients yielding

$$p_{k(j)}^{(1)}(\vec{r}) = c_0 + \sum_{i=1}^3 c_i x_i \quad (15a)$$

in second order ( $\sigma = 2$ ), we use quadratic polynomials with ten coefficients defined by the geometry of the considered tetrahedron  $\Omega_k^{(j)}$  (Figure 1)

$$p_{k(j)}^{(2)}(\vec{r}) = c_0 + \sum_{i=1}^3 c_i x_i + \sum_{i \geq 1}^3 c_{i,i} x_i x_i \quad (15b)$$



**Figure 1.** Left: first order  $\sigma = 1$  (linear approximation, eq 15a); the four vertices of the tetrahedron are used as control points. Right: second order  $\sigma = 2$  (quadratic approximation eq 15b); besides the four vertices of the tetrahedron also the six midpoints of the tetrahedron edges are used as control points yielding a total of ten control points.

Technical information about compiling the matrix elements  $A$  from the polynomial contributions of the different tetrahedrons can be found in refs 54 and 55.

It is known that a volume cannot be densely packed (without gaps) using perfectly regular tetrahedrons.<sup>56</sup> The art of volume meshing algorithm is to distort tetrahedrons as little as possible to obtain a dense packing obeying certain constraints such as the triangulated surface of a molecule. According to the results of volume meshing of different protein models with NETGEN<sup>42</sup> the average number of tetrahedrons per grid point is about 6 (see Table 1). In case of second order each tetrahedron carries an additional 6 grid points at the midpoint of the edges (Figure 1). Each of these additional grid points belongs on the average to about 1.5 tetrahedrons. In practical applications, we observed that the total number of grid points increased by a factor of about 7 going from first to second order solution. To compare the accuracy of electrostatic energy computations between FD and FE methods, we use the lattice constant for FD methods and the average distance of nearest neighbor grid points inside the protein volume ( $h_V$ ) for the present FE method. For first order approximation, this distance is equal to the average edge length of tetrahedrons in the protein volume. In the second order approximation, this distance is roughly equal to half the edge length, as explained above.

**C. Solving the Linear Poisson Equation Numerically.** To obtain the physically relevant solution of the electrostatic potential  $\Phi$ , we need to introduce appropriate boundary

**Table 1. Number of Tetrahedrons and Grid Points (Corner Points of Tetrahedrons) Inside and Outside of the Protein Lysozyme before and after Optimization Using  $h_S = h_V = 0.5 \text{ \AA}$ ,  $g = 0.5$**

volume	grid points	tetrahedrons
inside initial	39803	194921
inside optimized	38743	176874
outside initial	33035	155226
outside optimized	33981	158460
all initial <sup>a</sup>	55945	350147
all optimized <sup>b</sup>	55865	335526

<sup>a</sup>All initial is sum of inside and outside initial. Note that grid points on the protein surface ( $N_S = 16893$ ) are part of the inside and the outside set of grid points. In the sum, they are counted only once. <sup>b</sup>The whole volume model is subject to a final optimization step. Therefore, the numbers are smaller than the sum of individual contributions.

conditions, where the values of the electrostatic potential are known. This is the case for the grid points at the asymptotic boundary, where the electrostatic potential vanishes. We consider a known boundary value  $\Phi_i$  of the electrostatic potential at position  $\vec{r}_i$  by eliminating the corresponding  $i^{\text{th}}$  row and  $i^{\text{th}}$  column

$$\vec{a}_i = (\mathbf{A}_{1,i}, \mathbf{A}_{2,i}, \dots, \mathbf{A}_{i-1,i}, \mathbf{A}_{i+1,i}, \dots, \mathbf{A}_{N,i})^t$$

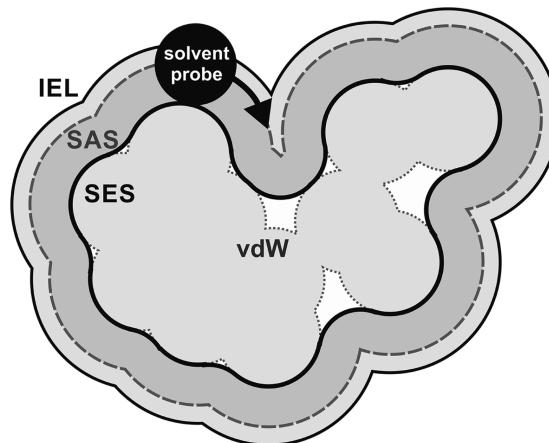
in the coefficient matrix  $\mathbf{A}$  of eq 13 and by replacing the right side vector  $\mathbf{b}$  by  $\mathbf{b}_i - \vec{a}_i \Phi_i$ , where  $\mathbf{b}_i$  is the vector  $\mathbf{b}$  with removed  $i^{\text{th}}$  component. This elimination procedure is performed for each known value of the electrostatic potential at the boundary. For vanishing electrostatic potential at the asymptotic boundary, the right side vector is just  $\mathbf{b}_i$ .

The resulting set of linear equations is sparse and small enough that it can be solved by iterative or even direct methods inverting the coefficient matrix numerically approximately or exact using the linear equation solver MUMPS (Multifrontal Massively Parallel sparse direct Solver).<sup>57–59</sup> MUMPS is specialized for sparse coefficient matrices and utilizes the symmetry of  $\mathbf{A}$ . If the coefficient matrix is inverted only approximately by MUMPS two iteration steps are enough to reach sufficient accuracy. For more details see results section.

In second order solution, the effective number of grid points increases by about the factor 7. If we decrease the average edge lengths of the tetrahedrons by a factor of 2 the number of grid points (in first order solution) formally increases by a factor of 8. Nevertheless, the accuracy of a first order solution of eq 13 with half the edge length is much lower than the accuracy of the second order solution with full edge length albeit the number of employed grid points are about the same. Hence, in second order solution the accuracy of the solution of eq 13 is improved not only by an increase in the number of grid points but also by replacing the linear interpolation [eq 15a] of the dielectric constant  $\epsilon(\vec{r})$  (defining the molecular surface) and the atomic point charge distribution  $\rho(\vec{r})$ , eq 2, by a quadratic interpolation according to eq 15b, which is performed for each tetrahedron. On a molecular surface meshed by triangles a linear interpolation represents the molecular surface by planar triangles. With quadratic interpolation the planar triangles are effectively replaced by curved surfaces, which can represent complex molecular surfaces more faithfully and thus increase the accuracy. However, in second order solution, the coefficient matrix  $\mathbf{A}$  is less sparse than in first order solution. As a consequence, the linear equation solver MUMPS<sup>57–59</sup> needs in

second order solution more iteration steps to solve the IPBE. In applications to proteins, one iteration is sufficient for first order and two iterations for second order solution. NETGEN can provide coefficient matrices and the right side vectors of linear IPBE, eq 13, also in higher than second order solution. In third and forth order, three iteration steps are necessary to get sufficiently converged results for the IPBE. In the present study, we generally employ second order solution with two iteration steps to solve the IPBE unless stated otherwise.

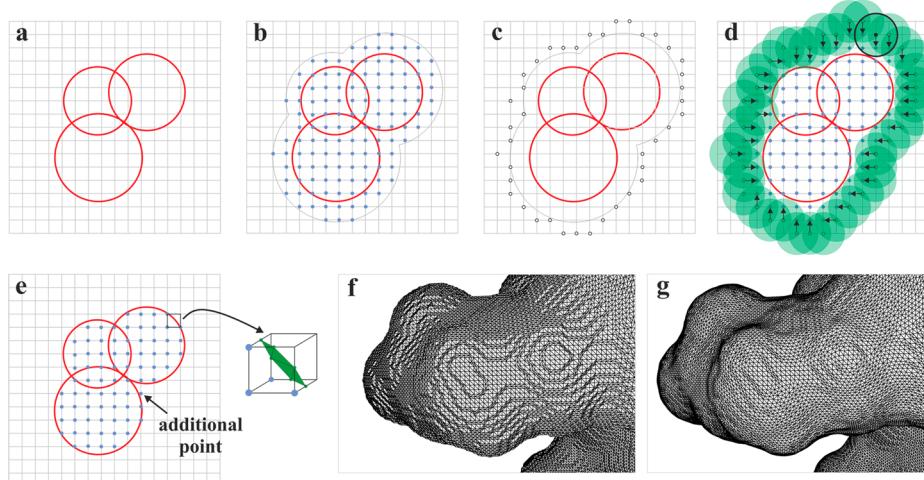
**D. Generating an Initial Molecular Surface.** To solve the IPBE for a molecule, we describe the molecular surface by triangles and cover the volume in and outside of the molecule by tetrahedrons. A classical definition of a molecular surface is due to Lee and Richards<sup>60</sup> in 1971. They defined the solvent accessible surface (SAS) of a molecule as the boundary of molecular volume, which is defined by the joint volume of all atoms described by their vdW spheres whose radii are enlarged by the probe sphere radius of the solvent molecules. Later, Richards<sup>61</sup> added a definition of molecular surface [later called solvent excluded surface (SES)],<sup>62,63</sup> which consists in a combination of contact and reentrant surface. In 1983, Connolly<sup>64</sup> demonstrated how one can calculate the molecular surface piecewise analytically. The description of molecular surfaces is complemented by an ion exclusion layer obtained by inflating the vdW spheres of the atoms by a counterion radius (Figure 2), which needs to be considered in case of



**Figure 2. Molecular surface definitions.** A solvent probe sphere (black sphere) rolls over the molecular surface defined by the boundary of the merged vdW volumes of all atoms of the molecule (molecular vdW surface). The solvent excluded surface (SES, inner black surface) of the molecule is given as the set of all contact points of the rolling sphere with the molecular vdW surface and the patches of the rolling sphere surface interpolating between the contact points to form a closed molecular surface. The solvent accessible surface (SAS, dashed surface) of a molecule is defined by the set of midpoints of the rolling sphere. The ion exclusion layer, also called Stern layer, is constructed by inflating the vdW radii by a counterion radius.

nonvanishing ionic strength  $\kappa^2$ . In the present study, we use the SES according to Richards<sup>61</sup> and Connolly,<sup>64</sup> which is commonly used for protein electrostatics.<sup>65</sup>

To obtain a high quality and robust initial description of a molecular surface with triangles, the triangles should have approximately equal edge lengths. For this purpose, we tried different meshing programs such as MSMS (Michael Sanner's Molecular Surface),<sup>63</sup> LSMS (Level-Set method for Molecular Surface generation),<sup>52</sup> EDTSurf (Euclidean Distance Transform



**Figure 3.** Initial template of the molecular surface considering a schematic model. (a) Resize the vdW volumes of all atoms of the molecule to fill a given simple cubic grid with lattice constant of 1 Å, where  $n = 512$  is number of lattice points in one dimension. (b) Enlarge the vdW spheres by adding a rescaled probe radius (1.4 Å for water). All lattice points inside the volume of the spheres are marked as being inside the molecule (blue points). (c) Find the outside lattice points (small open circles, contact points), which are in contact with the molecular surface. (d) The level-set method is applied, where solvent probe spheres (green, 1.4 Å radius) are placed in the contact points. Starting from the contact points one moves toward the nearest lattice point (blue points in b) inside the molecule and stops before one crosses the boundary of the probe sphere. The resulting set of terminal lattice points approximate the SES of the molecule. At interstitials of the vdW spheres additional points outside of the vdW volume can become part of the SES molecular volume to smoothen the surface (see part e). (e) The marching cubes algorithm<sup>73</sup> is applied to approximate the SES by triangles. (f) Molecular surface [part of lysozyme (PDB id 2LZT)]<sup>45</sup> obtained with LSMS.<sup>52</sup> (g) Molecular surface (same as in f) after 30 Taubin<sup>78</sup> smoothing operations used as input for NETGEN,<sup>42</sup> which generates a simplified, regularized triangular grid that represents the molecular SES.

Surf),<sup>66</sup> TetGen (Tetrahedral Mesh Generator),<sup>34</sup> as well as the ball-pivoting algorithm<sup>67</sup> in an own program applied to a point cloud on the molecular surface. Another alternative would be to use the program TMSmesh (Tracing Molecular Surface for meshing),<sup>68</sup> which was demonstrated to generate robust surfaces for BE and FE approaches. TMSmesh uses a Gaussian definition for the molecular surface that helps to produce robust results, but this procedure may lead to small interatomic cavities, which can cause artifacts.<sup>32</sup> Therefore, we have not considered this approach.

MSMS<sup>63</sup> is based on the analytical approach of the Connolly<sup>64</sup> surface. It is fast and represents molecular surfaces by triangles accurately with high point densities.<sup>69</sup> Therefore, it is often considered as reference surface and for graphic display.<sup>70,71</sup> At rims between reentrant and contact surface patches (appearing close to intersections of vdW spheres) the molecular surface triangles generated by MSMS can be very slim. A surface triangulation with such low quality triangles is difficult to repair. Their usage in FE methods usually leads to singular behavior of the resulting linear equations, eq 13. The analytical algorithm on which MSMS is based encounters singularities, if the SES exhibits self-intersecting parts leading for instance to cusp-like extrusions.<sup>72</sup> They may occur, if pairs of nearest neighbor atoms are at distances corresponding to the sum of their vdW radii and the diameter of the probe sphere. Such scenarios appear, for instance, at the entrance of cavities at a molecular surface whose opening diameter is twice the probe sphere radius. If MSMS detects such a singularity, the corresponding atomic vdW radii are enlarged by 0.1 Å, which closes the cavity entrance.<sup>63</sup> This manipulation leads to changes of the molecular surface, which locally can be significant, as we will see later.

EDTSurf<sup>66</sup> employs the marching cubes algorithm<sup>73</sup> to discriminate between inside and outside of a molecule. However, it uses only 23 out of  $2^8 = 256$  possible patterns,

which has led to artifacts. TetGen<sup>34</sup> produces a fine-grained meshing of a protein surface but has a tendency to yield coarse-grained tetrahedrons for the protein volume, where a refinement is difficult. Decherchi et al. propose an alternative molecular surface, called skin surface,<sup>30</sup> which is based on weighted Voronoi diagrams. This type of surface seems to be more favorable than the blobby surface. It is used in Delphi<sup>4</sup> to generate the molecular boundary within a simple cubic grid approach. Alternatively, Decherchi et al. developed the program NanoShaper,<sup>74</sup> which can also be used to generate molecular surfaces. It uses the CGAL library<sup>75</sup> and is integrated in Delphi.<sup>4</sup>

In the present study, we use LSMS,<sup>52</sup> which starts from a fine-grained simple cubic grid. LSMS<sup>52</sup> uses the level-set method<sup>76</sup> combined with the marching cubes algorithm.<sup>73</sup> We use LSMS<sup>52</sup> instead of the Connolly algorithm,<sup>64</sup> since with LSMS<sup>52</sup> we obtain molecular surface points suitable for a triangulation of the surface. Xu et al.<sup>66</sup> showed that for protein surfaces generated with LSMS<sup>52</sup> the volume error is smaller than 1%, if  $256^3$  lattice points are used. For our applications, computing electrostatic energies of proteins LSMS<sup>52</sup> reaches the best balance among accuracy, speed, and robustness, if  $512^3$  lattice points are used. LSMS<sup>52</sup> uses a fast version of the level-set method<sup>76</sup> to generate the molecular surface on an initial fine-grained simple cubic lattice (Figure 3). It applies the marching cubes algorithm<sup>73</sup> that uses all 256 possibilities of meshing cubes. Can et al.<sup>52</sup> showed that LSMS is capable of generating cavities in proteins, which are of comparable quality to the Swiss PDB viewer.<sup>77</sup> Since our mFES approach is based on LSMS, we are also able to add this feature in a future mFES version. We are using a slightly modified version of LSMS<sup>52</sup> to avoid accumulation points in NETGEN<sup>42</sup> giving rise to singular behavior. The level-set method is used from inside toward the vdW spheres (outward propagation), from outside to inside (first inward propagation) to generate the molecular surface, namely the SES, and from outside to inside again (second inward propagation) to

distinguish between protein and cavities, which are able to appear inside the protein.

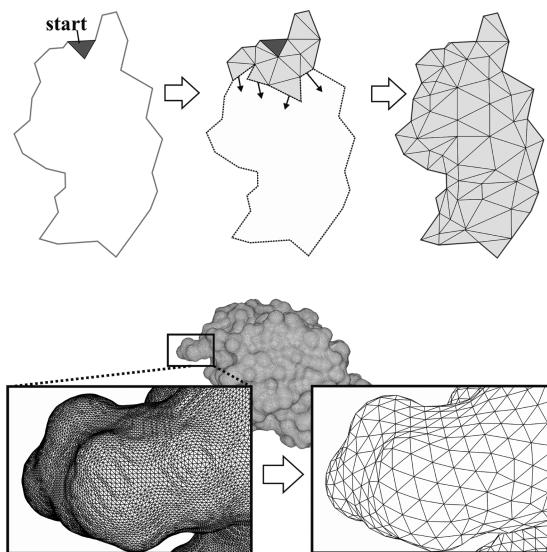
We modify the first inward propagation step toward the molecular surface (Figure 3), ignore singular needle-like spots on the molecular surface and apply the marching cubes algorithm,<sup>73</sup> which yields the molecular surface defined by a suitable set of grid points. However, this surface may be too corrugated and is therefore flattened by the Taubin<sup>78</sup> algorithm, which performs a two-step Gaussian<sup>79</sup> smoothing with one positive ( $\lambda = 0.50$ ) and one negative ( $\mu = -0.53$ ) smooth direction. This algorithm acts as a low-pass filter. It is known to generate well smoothed surfaces maintaining curvature faithfully and avoiding volume shrinkage. However, triangular self-intersections may occur with exceeding Taubin smoothing. Therefore, the number of Taubin smoothing steps was limited to 30 in the present application. As a result, we obtain a complete and faithful, but at the same time a sufficiently smooth, representation of the molecular surface by triangles. Unfortunately, the surface involves too many grid points rendering the meshing of volume inside and outside of the molecule difficult and computationally expensive. At this point, NETGEN<sup>42</sup> can be used to simplify the molecular surface representation without loss of quality and to mesh the volume inside and outside of the considered molecule as explained in the next section.

**E. Molecular Surface Meshing with NETGEN.** NETGEN<sup>42</sup> is a powerful and very efficient tool to mesh surfaces of all types of objects faithfully. It is particularly suited to model edges, narrow valleys, needles, and similar demanding structures in great detail. The surfaces of macromolecules such as proteins involve many such structural features in particular at the intersection of vdW spheres. We found that NETGEN has a tendency to pay too much attention to such details. In our application, we are interested in smooth molecular surfaces. Since vdW surfaces, which are used by some groups,<sup>82</sup> involve singular structural features, they cannot be handled well with mFES. However, the vdW surface of a molecule can be used after it is smoothed by a sphere of small radius that rolls over the surface. In our applications, we generally use a rolling sphere whose radius reflects the size of solvent molecules, which is 1.4 Å for water.

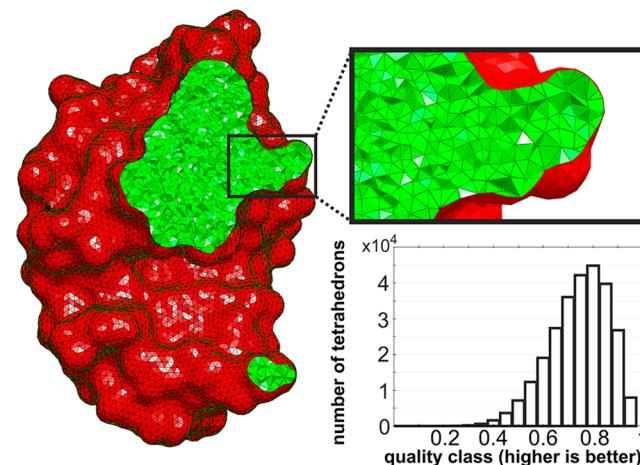
Starting from the initial template molecular surface characterized by grid points as described above, an advancing front method<sup>80,81</sup> is applied, which meshes the molecular surface with approximately equilateral triangles of a given edge length  $h_s$ . For this purpose, NETGEN<sup>42</sup> uses abstract rules to propagate from a suitable starting triangle until the whole molecular surface is covered with triangles (Figure 4).

**F. Volume Meshing and Construction of Spatial Model with NETGEN.** To solve the *l*PBE a complete spatial model needs to be constructed using the advancing front method of NETGEN.<sup>42</sup> First the volume inside the protein is meshed by tetrahedrons, starting from the triangles of the molecular surface (described in part E). Thereby, a constant average edge length  $h_v$  is used, which should be larger or equal to the edge length  $h_s$  of the surface triangles. Subsequently, the protein volume mesh is optimized (Figure 5) as described in part G below keeping the triangles on the molecular surface invariant. The result is shown for the protein lysozyme<sup>45</sup> in Figure 5.

As next step, an asymptotic boundary surface is defined. The volume mesh to be generated outside of the molecule extends in space up to the asymptotic boundary surface, which in case of FE can be of arbitrary shape in contrast to FD based methods. Since we use vanishing electrostatic potential at the asymptotic boundary, this boundary surface must be sufficiently distant

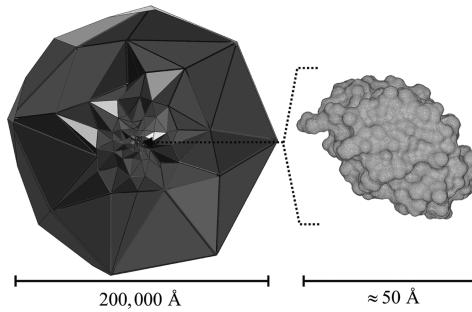


**Figure 4.** Advancing front method<sup>80,81</sup> is used to simplify and regularize the initial template of a triangular molecular SES given by the procedure described in Figure 3. (top) Starting from an arbitrary triangle of the given mesh (dark gray) a new more regularized and simplified mesh is grown on the molecular surface that uses the grid points of the initial mesh (shown schematically). (bottom) Nose structure of lysozyme with a foreground close-up; (bottom-left) initial template molecular surface with a high point density. (bottom-right) Molecular surface after applying the advancing front method with NETGEN.<sup>42</sup> We use an average edge length  $h_s = 0.5 \text{ \AA}$ , which results into a conserved molecular surface shape with less molecular surface points.



**Figure 5.** Molecular surface of lysozyme<sup>45</sup> was covered by triangles using the advancing front method of NETGEN<sup>42</sup> with an average edge size of  $h_s = 0.5 \text{ \AA}$ . The resulting protein surface involves 16893 grid points. The triangles on the protein surface are used as bases for tetrahedrons to perform the volume mesh inside the protein with an average edge length of  $h_v = 0.5 \text{ \AA}$ . The molecular surface is colored red. The molecular volume, shown as a planar cut, is colored green. The green nose on the lower right corresponds to the structural detail shown in Figures 3 and 4. The diagram with the distribution of the tetrahedrons in the different quality classes is explained later.

from the atomic point charges of the considered molecule. We use the surface of a sphere with radius of 100 000 Å as boundary and place the molecule in the sphere center (Figure 6). NETGEN<sup>42</sup> generates a coarse-grained mesh of triangles on the spherical boundary surface involving 23 grid points. NETGEN



**Figure 6.** FE model setup for the asymptotic boundary. The protein (e.g., lysozyme, right part) is placed in the center of a large sphere of radius 100 000 Å. The sphere possesses a coarse-grained surface mesh with 23 grid points corresponding to an average triangle edge length of about 80 000 Å at the boundary surface. The average triangle edge length on the protein surface is  $h_S = 0.5 \text{ \AA}$ , corresponding to 16893 grid points. NETGEN<sup>42</sup> covers the volume between the asymptotic boundary and protein surface with tetrahedrons with the advancing front method starting from the asymptotic boundary surface. Thereby, a grading parameter of  $g = 0.5$  is used, which reduces the edge lengths of adjacent tetrahedrons by a factor of 2, while they approach the protein surface. Thus, the distance between the asymptotic boundary and protein surface is on the average spanned by about  $n = \ln(80\,000)/\ln(2) \approx 16$  adjacent tetrahedrons. To turn off the influence of the parameter  $h_V$  for volume meshing outside of the protein, its value was set to  $10^6 \text{ \AA}$ .

starts from the boundary surface using the triangles at boundary surface as bases for tetrahedrons, to mesh the volume outside of the molecule. While propagating with the advancing front method from the boundary surface toward the molecular surface NETGEN<sup>42</sup> adjusts the size of the newly generated tetrahedrons to finally match with the fine-grained triangles on the molecular surface. The adjustment of edge length in this volume meshing procedure is controlled by the so-called grading parameter  $g$ ,<sup>42</sup> which provides the relative change of the edge lengths of neighbor tetrahedrons (generated with the advancing front method), which are oriented toward the molecular surface. Also, the volume mesh outside of the molecule is optimized keeping the triangles on the molecular surface invariant, as described in part G below.

Directly after applying the advancing front method for volume meshing NETGEN<sup>42</sup> routinely applies already a simple optimization of the generated tetrahedrons even if no explicit optimization step was requested keeping the asymptotic and molecular surface boundaries invariant. One or more extensive optimization cycles of the volume mesh are necessary and can be set up without changing the molecular surface. Finally, the whole volume model is optimized again to cure some problems, which may have been overlooked before. The separate generation of volume mesh in and outside of the molecule is crucial to keep the molecular surface invariant. The 3-fold iterative volume mesh optimization is important to obtain a well-defined grid for the molecular model, which guarantees a nonsingular behavior of the resulting linear equation system.

Based on simple test computations for Born ion electrostatic solvation energy, we found that for an accuracy of three significant digits a radius of 10 000 Å would also be sufficient. However, we prefer to use the larger sphere radius to be on the safe side. The volume mesh near the asymptotic boundary is very coarse-grained resulting in a very low density of grid points, which effectively does not enlarge the set of linear equations, eq 13.

**G. Optimizing Surface and Volume Mesh with NETGEN.** For the mesh described above, we evaluate for each tetrahedron  $k$  the dimensionless pseudo-energy-function

$$E_{\text{volume}}(\Omega_k) = \frac{1}{6^4 \sqrt{2} V_k} \left( \sum_{i=1,6} l_i^{(k)} \right)^3 + \sum_{i=1,6} \left( \frac{l_i^{(k)}}{h_V} + \frac{h_V}{l_i^{(k)}} - 2 \right) \quad (16a)$$

where  $V_k$  is the volume, while  $l_i^{(k)}$ ,  $i = 1, 2, \dots, 6$ , and  $h_V$  are the actual and desired edge lengths of tetrahedron  $k$ . An analog pseudo-energy-function can be evaluated for a triangle  $k$  of the molecular surface mesh

$$E_{\text{surface}}(\Gamma_k) = \frac{\sqrt{3}}{36} \frac{(\sum_{i=1,3} l_i^{(k)})^2}{S_k} + \sum_{i=1,3} \left( \frac{l_i^{(k)}}{h_S} + \frac{h_S}{l_i^{(k)}} - 2 \right) \quad (16b)$$

where  $S_k$  is the surface area of the triangle, while  $l_i^{(k)}$ ,  $i = 1, 2, \dots, 6$ , and  $h_S$  are the actual and desired edge lengths of triangle  $k$  on the surface, respectively. However, in the present applications, we do not optimize the surface mesh further on, since it is not critical for results but would alter the molecular surface. To optimize the volume meshes, we minimize the sum of the volume energy terms of all tetrahedrons

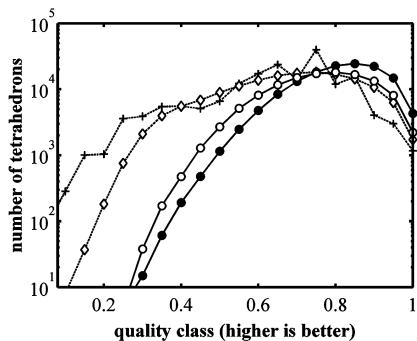
$$E_{\text{volume}}(\Omega) = \min \sum_k E_{\text{volume}}(\Omega_k) \quad (17)$$

by altering the volume mesh with tetrahedron swap, split, and collapse operations,<sup>42,83,84</sup> as explained in Figure S1 of the Supporting Information. These operations are very important, since without them the linear equation solver can become unstable.

To estimate the improvement of meshing by these optimization procedures NETGEN<sup>42</sup> uses the concept of quality classes to characterize the geometry of tetrahedrons. The corresponding quality parameter varies from 0 to 1, where zero refers to pure and unity to perfectly regular tetrahedrons. The distribution of tetrahedrons over the quality classes before and after optimization of the volume mesh inside and outside of the protein is shown for lysozyme in Figure 7 and summarized in Table 1. As one can see, there is a considerable improvement of tetrahedron quality and reduction of the number of grid points by the optimization procedure.

## RESULTS AND DISCUSSION

**General Overview.** In part A, we consider the Born ion model, compute the electrostatic solvation energies with nonvanishing ionic strength using different solvers, and compare the results with the analytical solution. In part B, we compute electrostatic solvation energies for four proteins of different size (bpti, barnase, lysozyme, CcO) and compare the results from mFES and FD solvers. Additionally, a comparison of mFES with other FE solvers is performed showing results on electrostatic solvation energy and differences in the representation of the protein surface for bpti. In part C, we provide detailed insight into mFES meshing parameters and their influence on the computed electrostatic solvation energies for lysozyme. In part D, we describe how the CPU time varies with the quality of



**Figure 7.** Number of tetrahedrons in the different quality classes according to NETGEN<sup>42</sup> based on lysozyme triangular surface meshing with average edge length of  $h_S = 0.5 \text{ \AA}$ . The initial volume meshes inside and outside of the lysozyme are generated with the advancing front method (dashed lines) using the triangular surface mesh. Inside of lysozyme (average edge length  $h_V = 0.5 \text{ \AA}$ ) data are given by +. Outside of lysozyme (volume edge lengths are adjusted to the edge lengths at the two boundary surfaces, which are protein surface and asymptotic boundary as described in Figure 6), the dependence on the quality class is given by ◊. The corresponding optimized results are displayed with solid lines (inside lysozyme ●; outside lysozyme ○).

generated meshes for different proteins. Finally, in part E, we show electrostatic potential maps generated for lysozyme and an adenovirus protein.

**A. Born Ion Model with Ionic Strength: Modeling One Ion As a Sphere.** For a proof of principle, we reproduce results for the Born ion model, which can be solved analytically. We consider a sphere of radius  $r_{\text{Born}} = 3 \text{ \AA}$ , with a unit charge  $q = 1 \text{ C}$  in the center and different dielectric constants inside ( $\epsilon_{\text{in}} = 4$ ) and outside ( $\epsilon_{\text{out}} = 80$ ) of the sphere. Furthermore, we introduce ionic strength described by the inverse Debye length  $\kappa$  outside of the Stern layer

$$\kappa^2 = \frac{8\pi e_0^2 I}{k_B T}, \quad I = \frac{1}{2} \sum_i c_i^{\text{bulk}} z_i^2 \quad (18)$$

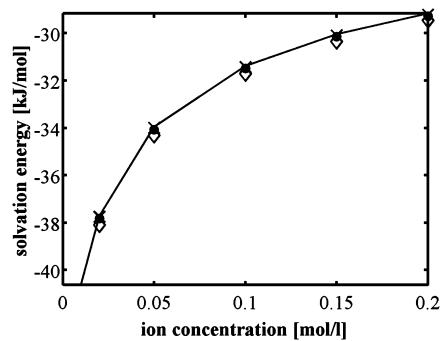
The constants in eq 18 are  $e_0$  elementary charge,  $k_B$  Boltzmann constant,  $T$  absolute temperature,  $c_i^{\text{bulk}}$  bulk ion concentration (number of ions of type  $i$  per unit volume), and  $z_i$  net charge of ion  $i$  in units of the elementary charge. The ionic strength  $I$  is a measure of the ion concentration in the solvent.

The Stern layer is defined by a rolling sphere whose radius  $r_{\text{ion}}$  is given by the type of ion. As a consequence the Stern layer is the surface of a sphere of radius  $r_{\text{Stern}} = r_{\text{ion}} + r_{\text{Born}}$  centered at the point charge  $q$ . The analytical form of the Born electrostatic solvation energy reads<sup>35,85,86</sup>

$$\Delta G_{\text{Born}} = \frac{q^2}{8\pi\epsilon_{\text{out}} r_{\text{Born}}} \left[ 1 - \left( 1 + \frac{r_{\text{Stern}}\kappa}{\sqrt{\epsilon_{\text{out}}}} \right)^{-1} \frac{r_{\text{Born}}\kappa}{\sqrt{\epsilon_{\text{out}}}} \right] - \frac{q^2}{8\pi\epsilon_{\text{in}} r_{\text{Born}}} \left[ 1 - \left( 1 + \frac{r_{\text{Stern}}\kappa}{\sqrt{\epsilon_{\text{in}}}} \right)^{-1} \frac{r_{\text{Born}}\kappa}{\sqrt{\epsilon_{\text{in}}}} \right] \quad (19)$$

At vanishing ionic strength ( $\kappa = 0$ ) the Born ion electrostatic solvation energy is  $\Delta G_{\text{Born}} = -54.995287 \text{ [kJ/mol]}$  for the given parameters, using values of the elementary constants from NIST<sup>87</sup> (see Supporting Information, Table S1). Analytical results using different ionic strength  $I$  are shown in Figure 8.

FD solvers of the  $\text{IPBE}$  equation use a simple cubic grid. To reach with such a grid the asymptotic limit for molecules, where



**Figure 8.** Ion electrostatic solvation energy of the Born model as a function of ion concentration  $I$ . Analytic solution, eq 19 (solid line), APBS<sup>3</sup> (coarse ◇; fine ×) and mFES (fine, ●) with ion exclusion layer using  $r_{\text{Born}} = 3 \text{ \AA}$ ,  $\epsilon_{\text{in}} = 4$ ,  $\epsilon_{\text{out}} = 80$ , and  $r_{\text{ion}} = 2 \text{ \AA}$ . The resolutions (fine and coarse) used for APBS and mFES are defined in the footnotes of Table 2. Detailed values are given in Supporting Information Table S2.

the electrostatic potential assumes known values, a focusing procedure needs to be applied. First, a large low resolution grid is used that reaches the asymptotic boundary. A grid with higher resolution is embedded in the low resolution grid. The boundary values of the high resolution grid are determined from the results of the low resolution grid. For proteins, a system of three or even more of such embedded grids are used.<sup>3,50,51,88</sup> To keep the number of focusing steps small FD solvers usually use the Debye–Hückel<sup>89</sup> approximation, for the electrostatic potential of a molecule at the asymptotic boundary,<sup>89</sup> which in the simplest case reads

$$\Phi(r) = \frac{q}{4\pi\epsilon_{\text{out}} r} \exp \left[ (b - r) \frac{\kappa}{\sqrt{\epsilon_{\text{out}}}} \right] \left( 1 + \frac{b\kappa}{\sqrt{\epsilon_{\text{out}}}} \right)^{-1} \quad (20)$$

where  $q$  is the total net charge of the molecule,  $\kappa$  the Debye length describing the ionic strength, and  $b$  the effective radius of the molecule including the Stern layer. mFES ignores the surface integrals, eqs 11, in the linear equation system eq 12. Thus, it needs to use a boundary surface that is very distant from the considered molecule, where the surface integrals and the electrostatic potential vanish. Therefore, mFES does not make use of the Debye–Hückel<sup>89</sup> approximation.

For the Born ion model the Debye–Hückel approximation for the asymptotic boundary, eq 20, is identical to the exact solution outside of the ion sphere of the Born model. Hence, the Born ion model is not an appropriate test case to study how the numerical solution of the  $\text{IPBE}$  depends on the boundary values of the electrostatic potential for the FD method if the Debye–Hückel approximation is used. Therefore, in the comparison between FD and FE using the Born ion model, we do not apply focusing steps for the FD methods (Table 2). However, such focusing steps will be necessary for molecular models to guarantee a high level of accuracy, since the Debye–Hückel approximation is not exact in such cases. We obtain high accuracy for electrostatic solvation energy of the Born ion model with all three considered solvers, APBS, MEAD, mFES (Table 2). Nonetheless, mFES is capable of solving the Born ion model for vanishing and finite ionic strength accurately, although mFES can use a much smaller number of grid points for the computation.

**B. Electrostatic Solvation Energy Calculations for Proteins. Comparison with FD Methods for Proteins of Different Sizes.** We compute the electrostatic solvation energies of four proteins

**Table 2.**  $\Delta G_{\text{Born}}$  Electrostatic Solvation Energy of a Unit Charge in Center of Sphere of Radius  $r_{\text{Born}} = 3 \text{ \AA}$ ,  $\epsilon_{\text{in}} = 4$ ,  $\epsilon_{\text{out}} = 80^a$

	accuracy	$\Delta G_{\text{Born}}$ [kJ/mol]	error [%]
APBS <sup>b</sup>	fine <sup>c</sup>	-55.0159	-0.037
	coarse <sup>d</sup>	-55.2722	-0.495
MEAD	fine <sup>c</sup>	no conv. <sup>f</sup>	
	coarse <sup>d</sup>	-54.7728	0.405
mFES	fine <sup>e</sup>	-54.9917	0.0065

<sup>a</sup>Comparison of FD and mFES solvers with vanishing ionic strength  $I = 0$ . <sup>b</sup>The point density at the atomic vdW spheres is set to 10 points/ $\text{\AA}^2$ , which is the recommended value in APBS. <sup>c</sup> $n^3 = 193^3 = 7.2 \times 10^6$  grid points with  $0.05 \text{ \AA}$  lattice constant. <sup>d</sup> $n^3 = 65^3 = 2.7 \times 10^5$  grid points with  $0.25 \text{ \AA}$  lattice constant. <sup>e</sup>Second order approximation is used corresponding to an average distance between neighbor grid points of  $0.175 \text{ \AA}$  inside the Born ion sphere resulting in a total of 34,335 grid points, which is 1/8 of the number grid points used for the coarse resolution with FD. The spherical asymptotic boundary surface is at a distance of  $10^5 \text{ \AA}$  from the center. <sup>f</sup>solving the linear equation system with successive over-relaxation<sup>4,5</sup> did not converge.

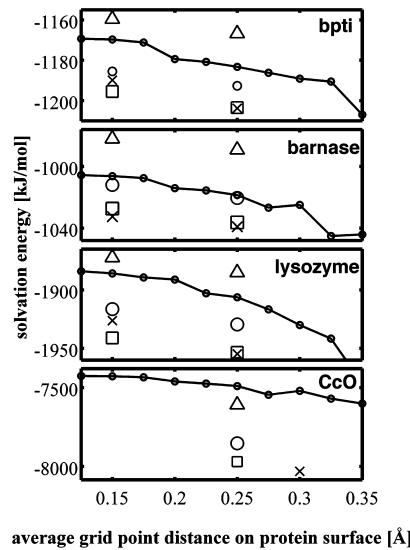
**Table 3. Four Proteins, Whose Electrostatic Solvation Energies Are Computed**

protein	PDB id	atoms <sup>a</sup>	(x, y, z) <sup>b</sup> [\text{\AA}]
bpti	1pit <sup>43</sup>	899	(27, 26, 45)
barnase	1a2p <sup>44</sup>	1724	(37, 39, 42)
lysozyme	2lzt <sup>45</sup>	1979	(33, 41, 49)
CcO	1m56 <sup>46</sup>	17868	(86, 82, 101)

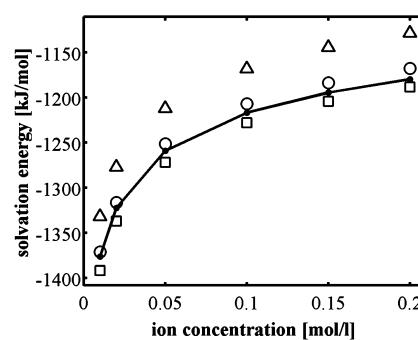
<sup>a</sup>Including hydrogen atoms; crystal waters are removed. <sup>b</sup>Diameter in x, y, z directions.

of different sizes with mFES (Table 3, protein structures are shown in Figure S2 of the Supporting Information) showing the change of electrostatic solvation energies using different average edge length for the triangles covering the protein surface (Figure 9). For these computations crystal waters were removed. As dielectric constant, we use  $\epsilon_{\text{protein}} = 4$  for protein volume and  $\epsilon_{\text{solvent}} = 80$  for the solvent in the implicit solvent model. Electrostatic solvation energies computed with mFES are plotted versus the average nearest neighbor grid point distance on the protein surface, while the edge length in the protein volume is kept constant at  $h_V = 0.5 \text{ \AA}$ . Since we apply the second order approximation, we consider also grid points at the edge midpoints of tetrahedrons decreasing the average distance of nearest neighbor grid points by nearly a factor of 2 as explained in the method part (C. Expanding the electrostatic potential with test functions). This factor of 2 is considered for the scale of the abscissa in Figure 9.

In Figure 9, we compare the electrostatic solvation energies of mFES with results from the FD methods of APBS<sup>3</sup> and MEAD.<sup>5</sup> In Figure 10, we show the influence of the ionic strength using the Stern layer for the lysozyme example. For both FD methods, we use a two-step focusing procedure. Focusing is automatically set up by APBS using the *mg-auto* parameter. In MEAD, the equivalent focusing scheme is set up manually. The number of grid points used in both focusing steps are the same as given in Table 4. In the second focusing step grid constants of 0.25 and  $0.15 \text{ \AA}$  are used for coarse and fine models, respectively. For the initial focusing step the FD methods use the Debye–Hückel<sup>89</sup> approximation for the asymptotic boundary to avoid the need for excessive focusing. The grid constant in the initial focusing step is given  $n$  (i.e., number of grid points in one Cartesian



**Figure 9.** Computed electrostatic solvation energies in units of [kJ/mol] for bpti, barnase, lysozyme, and CcO as a function of the average nearest neighbor grid point distance. Hydrogen atoms are added with CHARMM,<sup>90–92</sup> and their coordinates are energy minimized fixing all other atoms. We use atomic radii and partial charges from the CHARMM22<sup>91</sup> force field using standard protonation of the titratable residues. mFES: Solid line with dots, as a function of the average grid point distance which equals to two times the average edge length  $h_S$  of triangles in the molecular surface, since we use second order solution, which uses also grid points at the middle of the triangular edges. A constant edge length  $h_V = 0.5 \text{ \AA}$  in protein volume is used. APBS: Results are shown for coarse (second focusing step with  $h = 0.25 \text{ \AA}$ ) and fine (second focusing step with  $h = 0.15 \text{ \AA}$ ) resolution. For the construction of the molecular surface, three different point densities on the atomic vdW surfaces are considered: 10 points/ $\text{\AA}^2$  □; 3 points/ $\text{\AA}^2$  ○; 1 points/ $\text{\AA}^2$  Δ. MEAD: Results are marked with × and use same coarse and fine resolution for the second focusing step as does APBS. For CcO, we used  $0.3 \text{ \AA}$  for the coarse lattice constant because with  $0.25 \text{ \AA}$  no converged result could be obtained. For APBS and MEAD, CcO is too large to use a lattice constant smaller than  $0.25 \text{ \AA}$  where the lattice can still cover the whole CcO.



**Figure 10.** Example using lysozyme for computing ionic effect in solution with different solvers. mFES: Solid line with dots using  $h_S = 0.5 \text{ \AA}$ . APBS: Results for fine resolution (second focusing step with  $h = 0.15 \text{ \AA}$ ). For the construction of the molecular surface three densities on the atomic vdW surfaces of the atoms are considered: 10 points/ $\text{\AA}^2$  □; 3 points/ $\text{\AA}^2$  ○; 1 point/ $\text{\AA}^2$  Δ. Including the Stern layer in mFES increases the number of grid points by a factor of 2 resulting in a corresponding larger number of linear equations and therefore larger CPU time to solve the linear equation system. We believe this can be improved in future mFES releases.

**Table 4. Number of Linear Equations (dof) (Identical to Number of Grid Points) Used to Solve the PB Equation with APBS and mFES<sup>a</sup>**

proteins	APBS coarse	APBS fine	mFES
bpti	$n^3 = 193^3$	$n^3 = 321^3$	$2.3 \times 10^5$
	$7.2 \times 10^6$	$33 \times 10^6$	
barnase	$n^3 = 193^3$	$n^3 = 321^3$	$3.0 \times 10^5$
	$7.2 \times 10^6$	$33 \times 10^6$	
lysozyme	$n^3 = 225^3$	$n^3 = 353^3$	$3.5 \times 10^5$
	$11.4 \times 10^6$	$44 \times 10^6$	
CcO	$n^3 = 449^3$	not possible	$2.1 \times 10^6$
	$90.5 \times 10^6$		

<sup>a</sup>For APBS, we consider the number of dof only for one focusing step.  $n$  is the number of grid points along the Cartesian axes.

direction, see Table 4), determined such that the grid extension is four times as large as the maximum extension of the protein along the three Cartesian axes.

The dependence of electrostatic solvation energies on grid resolution obtained with mFES becomes constant at high resolution (Figure 9) (average nearest neighbor grid point distance at the protein surface below 0.2 Å) yielding a converged result. Hence, these electrostatic solvation energies can be considered to be precise. At lower resolution the electrostatic solvation energy is consistently overestimated, which corresponds to a molecular volume, which is underestimated. In the mFES method, the molecular surface is covered by planar triangular tiles. Since a molecular surface seen from the outside is predominantly convex, covering the surface with planar tiles, whose corner points are precisely on the surface, underestimates the molecular volume. This corresponds to the systematic trend observed in Figure 9 that the molecular electrostatic solvation energies become more negative at lower resolution. If a 1% precision for electrostatic solvation energy is sufficient, a neighbor grid point distance of 0.25 Å is enough. For second order Lagrange polynomial approximation this corresponds to an average edge length of the surface triangles of 0.5 Å.

APBS uses different point densities on the vdW spheres of the atoms, while it generates a molecular surface model on a simple cubic grid. The vdW sphere point density changes the surface of a molecular model, such that for high point densities (10 points/Å<sup>2</sup>) the molecular volume is smaller than for low point densities (1 points/Å<sup>2</sup>). As a consequence the electrostatic solvation energy is overestimated for large and underestimated for low point densities (Figure 9). Seemingly electrostatic solvation energies that agree best with the converged values from mFES are obtained for the low point density of 1 points/Å<sup>2</sup>, albeit the recommended value is 10 points/Å<sup>2</sup>. In the web application Karlsberg+<sup>50,51</sup> for pK<sub>A</sub> computations in proteins this point density is set to 3 points/Å<sup>2</sup>. Similar as with mFES the electrostatic solvation energies based on FD computations are less negative for higher grid resolution. However, the precision of the electrostatic solvation energy is not necessarily improving at higher simple cubic grid resolution (0.15 instead of 0.25 Å), since the limiting factors of modeling the molecular surface depends also on the atomic vdW sphere point density.

The slight variation of the molecular surface with different grid point densities on the atomic vdW surfaces used by APBS influences the electrostatic solvation energy for finite ionic strength in a similar way as for vanishing ionic strength as demonstrated in Figure 10. The results obtained with mFES are between the obtained values with APBS using a high (10 points/

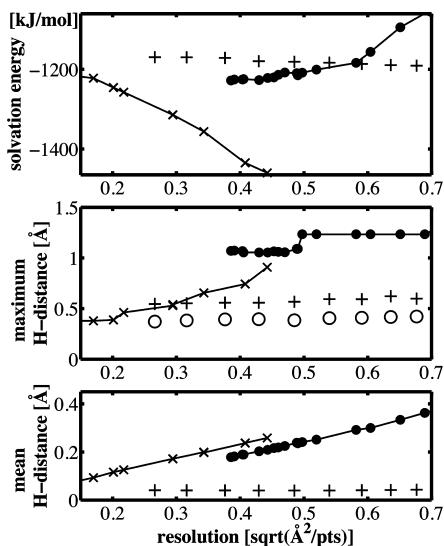
Å<sup>2</sup>) and a intermediate (3 points/Å<sup>2</sup>) grid point density for the atomic vdW spheres. A similar behavior is observed for the four proteins considered in Figure 9 and the discussion before.

MEAD uses an iterative SOR<sup>4</sup> method to solve the linear equation system of the PB equation. For larger molecules such as CcO or larger grids with small grid constants convergence with this method can be difficult. For CcO convergence with MEAD was not obtained even for a coarse grid with lattice constant 0.25 Å for the second focusing step. Using the slightly larger lattice constant of 0.30 Å the PB equation converged. These convergence problems occurred for vanishing ionic strength and may not occur for finite ionic strength, where the asymptotic values of the electrostatic potential vanish exponentially [see eq 20]. The result is shown in Figure 9. For both APBS and MEAD, it was not possible to compute the electrostatic solvation energy of CcO using a fine grid constant of 0.15 Å, since the cubic grid gets too large requiring far more than 128 GB of main memory. This is why CcO results, obtained with APBS, can only be shown for the coarse model with a grid constant of 0.25 Å (Figure 9).

**Influence of Cavities on the Electrostatic Solvation Energy.** The computed molecular electrostatic solvation energy may depend on cavities, which can be filled by solvent. The present version of mFES ignores internal not directly solvent accessible cavities; that is, the dielectric constant in such cavities is set to  $\epsilon_{\text{protein}} = 4$  like being filled by protein moiety. In a future release of mFES, we will implement also a cavity feature. APBS considers such cavities by setting the dielectric constant inside these cavities to  $\epsilon_{\text{solvent}} = 80$ . To explore the influence of such cavities on protein electrostatic solvation energy, the electrostatic solvation energy of lysozyme was also computed with APBS after filling cavities with zero charged small dummy atoms, whose dielectric constant is  $\epsilon_{\text{protein}} = 4$ . There was only a single cavity of about 30 Å<sup>3</sup>, which in the crystal structure hosts a single water molecule. Ignoring this cavity, the electrostatic solvation energy of lysozyme computed with APBS under the conditions of the high resolution fine model (for a definition see Table 4) yielded an electrostatic solvation energy, which is 3 [kJ/mol] less negative. The much smaller protein bpti possesses a chain of three crystal water molecules in an internal cavity.<sup>93</sup> Unfortunately, the cavity is too narrow and could therefore not be generated with APBS. There are larger internal cavities in CcO,<sup>94</sup> which is, however, too large to evaluate the electrostatic solvation energy with high resolution fine model. We do not expect that small cavities in proteins have a large influence on the value of electrostatic solvation energy to explain the discrepancies of the results obtained with FD and FE methods. For CcO, the internal cavities are larger and may therefore have some influence on the computed electrostatic solvation energy.

**Comparing mFES with Other FE Methods.** We now compare mFES with two other FE programs (PBF and AFEM) demonstrating results for the protein bpti<sup>43</sup> as a function of resolution. Since in PBF<sup>23,24</sup> a limit of 1000 atoms is set, we could not make a comparison for larger proteins. AFEM<sup>26</sup> is a finite element program implemented in APBS<sup>3</sup> and PBF<sup>23,24</sup> is part of Jaguar software package distributed by the Schrödinger molecular modeling company.<sup>22</sup> To perform an objective and fair comparison of the dependence on resolution, we evaluate the surface area of bpti with MSMS yielding  $a_{\text{bpti}} = 3197.44$  Å and count the number of grid points  $n_{\text{bpti}}$  representing the molecular surface and define  $(a_{\text{bpti}}/n_{\text{bpti}})^{1/2}$  as resolution. With mFES the electrostatic solvation energy of bpti converges to -1169.74 [kJ/mol] (Figure 9 and top of Figure

11). The AFEM electrostatic solvation energies tend to converge toward the mFES results with increasing resolution (decreasing average edge length of the molecular surface triangles) although higher resolution than with mFES is needed, top of Figure 11. The electrostatic solvation energies obtained with PBF appear to be converged at a different energy, namely,  $-1225$  [kJ/mol].



**Figure 11.** Comparing different FE solvers. (top) The electrostatic solvation energy [kJ/mol] of the protein bpti<sup>43</sup> is computed with the programs mFES (+), AFEM<sup>26</sup> available in APBS (x), and PBF<sup>23,24</sup> (●). An asymptotic boundary distance of  $100\,000$  Å is used for mFES, AFEM, and PBF. (middle) Hausdorff distance<sup>95,96</sup> (maximum H-distance) in units of [Å] of the molecular surfaces of bpti (evaluated with the program available in Meshlab<sup>97</sup>). As reference molecular surface, we use the surface generated with MSMS<sup>63</sup> using a resolution of  $10$  points per  $\text{\AA}^2$ . The circles (○) show the maximum H-distance, after removing an MSMS artifact as explained in text. (bottom) Mean H-distance. For the H-distances the same notation as in the top part is used.

To explore the quality of the molecular surfaces of bpti<sup>43</sup> generated with different FE methods, we measure the difference of these surfaces with the surface obtained with MSMS<sup>63</sup> evaluating the Hausdorff distance<sup>95,96</sup> (maximum H-distance) between these surfaces, which is defined as

$$d_{\max\text{-H}}^{\text{MSMS,FE}} = \max \left( \max_{j} \{ \min_i [d(\vec{r}_j^{\text{MSMS}}, \vec{r}_i^{\text{FE}})] \}, \right. \\ \left. \max_i \{ \min_j [d(\vec{r}_j^{\text{MSMS}}, \vec{r}_i^{\text{FE}})] \} \right) \quad (21)$$

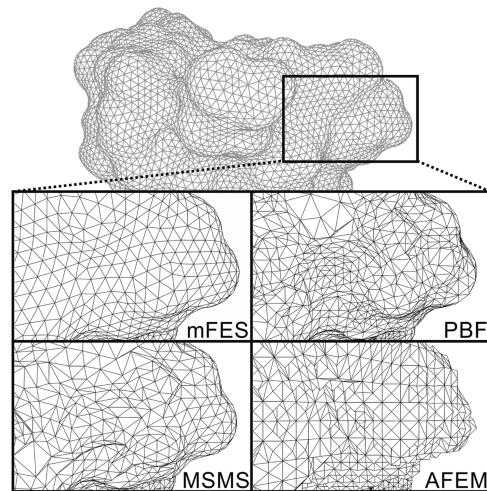
where  $\vec{r}_j^{\text{MSMS}}$  and  $\vec{r}_j^{\text{FE}}$  are the grid points on the surfaces generated with MSMS and one of the considered FE methods, respectively, and  $d$  is the distance function between a pair of grid points from different surfaces. The maximum H-distance between two equivalent surfaces effectively measures the maximum distance of equivalent pairs of points on the two surfaces and is thus a faithful measure of the maximum local deviation of two surfaces (Figure 11, middle part). Since a local deviation can be large, while the overall deviation of two equivalent surfaces may still be small, we also consider the average distance of all pairs of equivalent surface points that we call mean H-distance (Figure 11, bottom part) defined as

$$d_{\text{mean-H}}^{\text{MSMS,FE}} = \max \left( \text{mean}_{j} \{ \min_i [d(\vec{r}_j^{\text{MSMS}}, \vec{r}_i^{\text{FE}})] \}, \right. \\ \left. \text{mean}_i \{ \min_j [d(\vec{r}_j^{\text{MSMS}}, \vec{r}_i^{\text{FE}})] \} \right) \quad (22)$$

These two H-distances are shown as a function of resolution for surfaces of bpti generated with mFES, AFEM<sup>26</sup> and PBF<sup>23,24</sup> relative to the SES surface generated with MSMS<sup>63</sup> as a reference in Figure 11, middle and bottom part. To evaluate the H-distances Metro,<sup>96</sup> implemented in Meshlab,<sup>97</sup> is used.

The maximum H-distance between the reference SES of bpti obtained with MSMS<sup>63</sup> versus mFES is practically independent of resolution, but are with  $0.6$  Å (+, middle part of Figure 11) relatively large. The maximum H-distance of MSMS versus AFEM<sup>26</sup> seems to converge to  $0.4$  Å (x, middle part of Figure 11), which still is quite large. These differences are the result of a number of very localized but significant deviations in the MSMS surface, which are due to enlarged atomic vdW radii that are changed to avoid singularities in the underlying analytical algorithm,<sup>72</sup> as discussed in more detail in the method section D. To demonstrate this effect, we identified the surface location, which is responsible for this deviation between the bpti surface generated with MSMS and mFES. When we ignore the small number of grid points in the MSMS and mFES surface corresponding to this singular region the maximum H-distance decreases from  $0.6$  to  $0.4$  Å (open circles in middle part of Figure 11). The maximum H-distance of the surfaces generated with PBF and MSMS is considerably larger and shows some dependence on resolution used for the PBF surface but remains relatively large even at high resolution. A close-up of the bpti surface generated with PBF unravels some problems in comparison with the surface obtained by other methods (Figure 12).

The mean H-distance between the bpti surface from MSMS and the surfaces obtained with mFES is very small and shows practically no dependence on resolution in contrast to AFEM



**Figure 12.** Comparison of molecular surfaces (SES) generated with different FE methods, focusing on a small region of bpti.<sup>43</sup> The SES of the whole protein (top part) is generated with mFES using  $h_s = 0.5$  Å [ $0.54$  Å resolution measured as  $(a_{\text{bpti}}/n_{\text{bpti}})^{1/2}$  with  $a_{\text{bpti}}$  surface area of bpti and  $n_{\text{bpti}}$  number of grid points on surface]. Compared are mFES with MSMS<sup>63</sup> ( $4$  pts/ $\text{\AA}^2$  corresponding to  $0.5$  Å resolution), PBF<sup>23,24</sup> ( $0.39$  Å resolution), and AFEM<sup>26</sup> ( $0.44$  Å resolution).

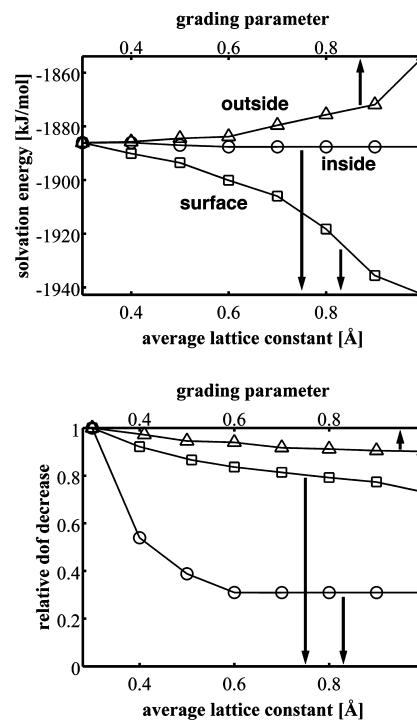
and PBF. The latter two show a monotonous decrease of the mean H-distance with increasing resolution but remain above the mean H-distance between the MSMS and mFES surface. The residual deviation between MSMS and mFES surfaces corresponds to a mean H-distance of about 0.045 Å. It is due to the underlying resolution used in LSMS<sup>52</sup> (512<sup>3</sup> simple cubic grid points) that provides the template for the triangular meshing of the molecular SES with NETGEN.<sup>42</sup> Since bpti has a maximum extension of 45 Å, the 512<sup>3</sup> simple cubic grid used with LSMS embeds the whole protein with a grid constant of 0.1 Å. Hence, the resolution for the molecular surface points is better than 0.05 Å in good agreement with the obtained residual mean H-distance. The smallness of the mean H-distance demonstrates that the large maximum H-distances between the molecular surfaces obtained with MSMS and mFES occur only at a number of very small regions involving narrow cavities that are ignored by MSMS. Therefore, we believe that the mean H-distance is a more reliable measure of deviations between molecular surfaces in particular if the reference surface to compare with involves small local but significant deviations as is the case for the MSMS surface.

To obtain an idea about quality, regularity, density, and shape of molecular surfaces, we compare the SES surfaces of the three considered FE solvers with the corresponding reference surface computed with MSMS<sup>63</sup> (Figure 12) considering a detail of the protein surface of bpti. The molecular shapes are very similar but differ in the regularity of triangles. The surface obtained with MSMS exhibits in several locations very slim triangles. These are likely the reason that our attempt to use a molecular surface triangulated with MSMS<sup>63</sup> as starting point of NETGEN<sup>42</sup> failed. Obviously, the molecular surface generated by mFES has a high quality and is very regular whereas other FE based surfaces face some problems (Figure 12). In addition mFES has a very low mean H-distance relative to the surface generated with MSMS (bottom part of Figure 11). The AFEM<sup>26</sup> surface shows still traces from the simple cubic lattice, which is used as a starting point to mesh the molecular surface with triangles. mFES uses initially with LSMS,<sup>52</sup> which also starts with a high resolution simple cubic lattice. NETGEN succeeds to regularize and simplify the initial meshing obtained with LSMS such that no traces of the initial cubic grid remain visible.

**C. Influence of Surface and Volume Meshing on Solvation Energies.** In this section, the most important parameters for surface and volume meshing provided by mFES will be discussed. Their influence can be judged by the quality class assignments of the final molecular model as provided by mFES. If there are few or zero elements in quality classes below 0.2 (Figures 5 and 7) an accurate result for the computed electrostatic solvation energy can be expected. If this condition is not fulfilled, one or more explicit volume optimization steps should be applied. mFES uses three main control parameters for this purpose: (i) average edge length of the molecular surface,  $h_s$ , (ii) average edge length of tetrahedrons in the protein volume  $h_v$ , and (iii) a grading parameter  $g$ , which controls the relative change of the edge lengths of neighbor tetrahedrons beginning at the spherical asymptotic boundary and moving toward the molecular surface in the center of the boundary sphere. A compromise between accuracy and CPU time for these three parameters in mFES is  $h_s = 0.5 \text{ \AA} = h_v$  and  $g = 0.5$ . (If not otherwise stated we use  $g = 0.5$ ) However, there are more ways to control surface and volume meshing in mFES. Normally, the user should only change the parameter of the edge length of the triangles on the molecular surface ( $h_s$ ) and

might change the number of Taubin steps (smoothing steps, default value  $t = 30$ ) performed to smooth the initial surface to obtain a robust meshing result. The interested reader is referred to the webpage at <http://agknapp.chemie.fu-berlin.de/mfes>.

According to mFES, the electrostatic solvation energy of lysozyme converges at a value of  $-1886.09 \text{ [kJ/mol]}$  (Figure 9 and Figure 13, top), while there is nearly no dependence on the

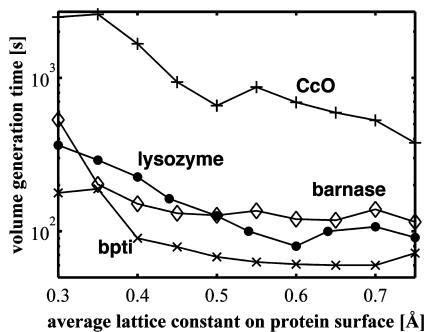


**Figure 13.** Meshing parameters of mFES influence the computed electrostatic solvation energy for lysozyme.<sup>45</sup> The plots show the dependence of relative electrostatic solvation energy (top) and the relative dof (bottom) (both referring to the converged result at high resolution) as a function of the effective average lattice constant ( $h_s$  □;  $h_v$  inside the protein ○) and the grading parameter ( $g$ , Δ, upper scale). Larger lattice constant and larger grading parameter corresponds to smaller number of dof (linear equations). The molecular models are obtained for the grading parameter  $g = 0.3$  and for protein surface and protein volume the average edge lengths  $h_s = 0.3 \text{ \AA} = h_v$ , while outside of the protein an edge length of  $10^6 \text{ \AA}$ , larger than the radius of the boundary sphere ( $h_v > 10^5 \text{ \AA}$ ), is used (see also Figure 6). (top) Electrostatic solvation energy as a function of average lattice constant for protein surface (□) and protein volume (inside, ○) and grading parameter influencing the grid point density outside of the protein (Δ). (bottom) Same plot for the relative decrease of the dof. The number of linear equations is reduced by nearly 70% if  $h_v \geq 0.55 \text{ \AA}$  inside the protein. The electrostatic solvation energy is influenced mostly by the average edge length of tetrahedrons on the molecular surface.

average edge length of tetrahedrons in the protein volume. A large grading parameter for the tetrahedrons outside of the protein renders electrostatic solvation energies less negative. The most influential parameter is the average edge length of the triangles on the molecular surface ( $h_s$ ). With smaller  $h_s$  the molecular surface resolution is finer, which generates a molecular surface, which is closer to the exact molecular surface definition (SES). For the FD method APBS the computed electrostatic solvation energy is between  $-1872$  and  $-1953 \text{ [kJ/mol]}$  and varies with the point density on the atomic vdW surfaces (Figure 9, lysozyme). In general, more accurate

electrostatic solvation energies using fewer linear equations are computed with mFES, mainly since it is able to adopt a high molecular surface resolution while using less dof.

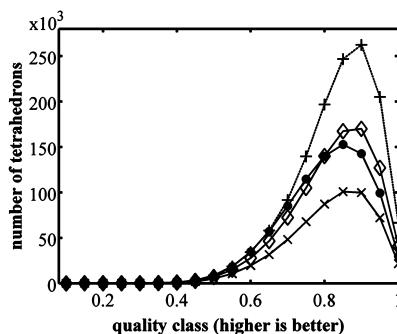
**D. CPU Times and Meshing Quality.** The number of linear equations (given by the number of grid points) to be solved for mFES grows essentially with the molecular surface area, which is approximately proportional to  $N_{\text{atom}}^{2/3}$ . Hence, for mFES the number of equations grows slower with the size of a molecular system than for FD methods where the number of grid points  $n^3$  is approximately proportional to the number of atoms ( $N_{\text{atom}}$ ). Therefore, solving the linear equation system of the PB equation obtained with mFES is fast (due to the relatively small number of grid points) and nevertheless very accurate. The required CPU times for this part (excluding the generation of the tetrahedral grid) are shown as a function of grid resolution in Figure S3 of Supporting Information. However, the generation of the molecular model (i.e., determining the grid points describing the molecular surface and covering the volume inside and outside of the molecule) to set up the linear equation system needs a considerable amount of CPU time. In contrast, the FD approach needs practically no time to generate the molecular model and to set up the linear equation system, since it is based on the simple cubic grid. However, the precision of the molecular surface representation with FD is somewhat limited. The CPU times needed for the model generation of bpti, barnase, lysozyme, and CcO using NETGEN<sup>42</sup> as a function of the lattice constant on the protein surface are given in Figure 14. The CPU time decreases only moderately, if the



**Figure 14.** CPU times [Intel Xeon X5690 with 3.47 GHz] needed by NETGEN<sup>42</sup> to mesh and optimize the tetrahedrons covering the volume in and outside of the protein as a function of the average edge length of molecular surface triangles  $h_s$ . The other meshing parameters are set to  $h_v = 0.5 \text{ \AA}$  inside,  $h_v = 10^6 \text{ \AA}$  outside protein volume and for the grading parameter  $g = 0.5$ . The CPU time to generate an initial molecular surface with LSMS<sup>52</sup> is not included. The corresponding CPU times for the LSMS<sup>52</sup> procedure, modified as described in text, are bpti ( $\approx 27 \text{ s}$ ), barnase ( $\approx 42 \text{ s}$ ), lysozyme ( $\approx 33 \text{ s}$ ), and CcO ( $\approx 44 \text{ s}$ ).

resolution is lowered by increasing the average edge lengths  $h_s$  of the triangles modeling the protein surface beyond  $0.5 \text{ \AA}$ , since the protein volume mesh is based on an average edge length of  $h_v = 0.5 \text{ \AA}$ .

In Figure 15, we show the distribution of tetrahedrons over the quality classes as defined by NETGEN<sup>42</sup> for the four considered proteins. For more details see part G. “Optimizing surface and volume mesh with NETGEN” in the method section. The total number of grid points and tetrahedrons generated for the four molecular models is given in Table 5. There should be practically no tetrahedrons associated with quality classes below 0.2 as is the case for the four considered proteins in Figure 15. Applying at the end a single optimization



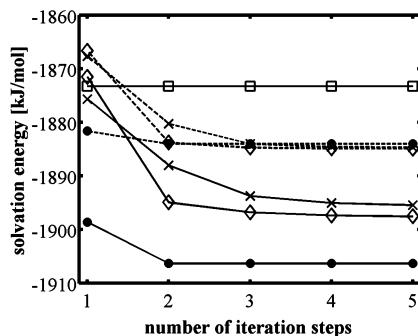
**Figure 15.** Quality class diagram of the whole protein model meshed with tetrahedrons after optimization with NETGEN<sup>42</sup> for bpti ( $\times$ ), barnase ( $\diamond$ ), lysozyme ( $\bullet$ ), and CcO (+) using an average edge length of  $h_s = 0.3 \text{ \AA}$  on the molecular surface. The quality of a tetrahedron is unity, if it possesses maximum volume for constant surface. For CcO, the number of tetrahedrons is decreased by a factor of 6 to fit into the figure (+). The total number of grid points and tetrahedrons are given in Table 5.

**Table 5. Resulting Number of Grid Points and Tetrahedrons for the Considered Proteins Using  $h_s = 0.3 \text{ \AA}$ ,  $h_v = 0.5 \text{ \AA}$  Inside,  $h_v = 10^6 \text{ \AA}$  Outside Protein Volume and for the Grading Parameter  $g = 0.5$**

protein	grid points	tetrahedrons
bpti	96714	567226
barnase	157558	922979
lysozyme	151900	888028
CcO	1359590	7993744

cycle for the inner and outer molecular volume as described in the method section part G, the quality classes distribution still improves and its maximum shifts to larger values near 0.85. If more optimization steps are done the number of tetrahedrons associated with high quality classes will rise up even more until an upper bound is reached, which is limited by the complexity of the considered molecule. However, it should be noted that each optimization cycle increases the CPU time to generate the molecular model.

Analyzing the convergence behavior of the electrostatic solvation energy of lysozyme (precise value  $-1886.09 \text{ [kJ/mol]}$ ) with the number of iteration steps to solve the LPBE with MUMPS<sup>57–59</sup> (Figure 16) we note that a convergence using second order solution for moderate resolution ( $h_s = 0.5 \text{ \AA}$ , corresponding to a resolution of  $0.25 \text{ \AA}$ ) does not yield a very precise electrostatic solvation energy as discussed for Figure 9 but is a good trade-off between accuracy and CPU time (Figure 14). Using the same molecular model but a higher solution order, the accuracy improves. A fine grained molecular model ( $h_s = 0.25 \text{ \AA}$ , corresponding to a resolution of  $0.125 \text{ \AA}$ ) yields very precise electrostatic solvation energies even with second order solution. There are practically no further improvements using a higher order solution with fine grained model. The first order solution converges after one iteration step, the second order solution after two iteration steps and the third and forth order solutions of approximately after three iteration steps solving the linear equations. This behavior correlates with the sparsity of the coefficient matrix A. For a less sparse coefficient matrix (larger bandwidth) the linear equation solver MUMPS<sup>57–59</sup> needs more iterations steps for convergence. The bandwidth of the coefficient matrix A is given by the number of points used for individual tetrahedrons [see Figure 1

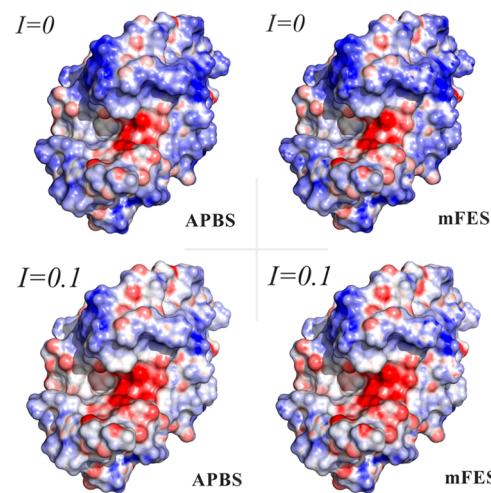


**Figure 16.** Convergence of the electrostatic solvation energy of lysozyme using different orders to solve the iPBE with mFES using two tetrahedral grid models [average molecular surface edge length of  $h_s = 0.5 \text{ \AA}$  (solid line) and  $h_s = 0.25 \text{ \AA}$  (dashed line)]. The number of linear equations is in second ( $\bullet$ ) order solution larger by about a factor 8 than in first ( $\square$ ) order solution. For third ( $\diamond$ ) and forth ( $\times$ ) order solution the number of equations increase relative to first order solution by about a factor of 26 and 64, respectively. Since the coefficient matrix of the linear equation system is sparse, CPU time to solve the iPBE is proportional to the number of equations to be solved. A convergence for second order solution is reached after two iteration steps to solve the linear equations. For third and forth order solution three iteration steps are needed for convergence.

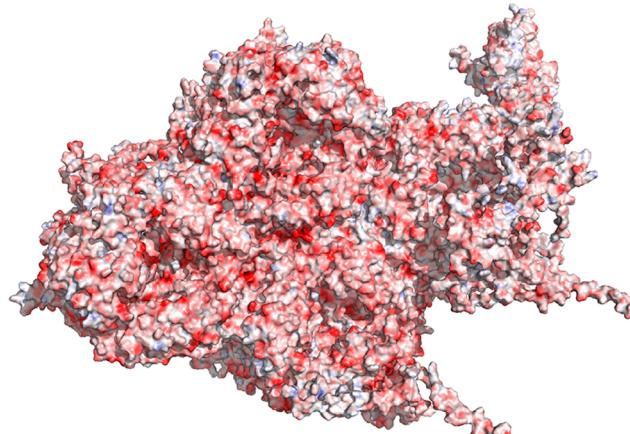
and eq 15b]. The CPU time solving the linear equation system in second order solution increases by a factor of 4 using a high resolution ( $h_s = 0.25 \text{ \AA}$ ) rather than a moderate resolution ( $h_s = 0.5 \text{ \AA}$ ) model. On the other hand, comparing first and second order solution of the same molecular model, we observe an increase in CPU time by a factor of 8. Note that the increase in CPU time is larger in this case, since in second order solution two iteration steps are needed to obtain convergence, while one iteration step is sufficient for first order solution.

**E. Electrostatic Potential Maps.** mFES is capable to provide electrostatic potential data for a molecule, which can be mapped on its surface. These potential maps can be viewed by programs such as VMD<sup>70</sup> or PyMOL.<sup>98</sup> The visualization programs need such data on a simple cubic grid. Since mFES works on an irregular tetrahedral grid, the necessary values on the simple cubic grid are obtained by interpolation. Figure 17 shows such electrostatic potential maps generated by APBS and mFES. Visually, there is nearly no difference between the two programs. Naturally, one could also use NETGEN<sup>42</sup> directly to visualize the electrostatic potential using the irregular grid points.

Finally, we generate the electrostatic potential map for an adenovirus protein (PDB id 4CWU<sup>47</sup>) involving 193 000 atoms (including hydrogen atoms) displayed with PyMOL<sup>98</sup> in Figure 18. For this protein an electrostatic solvation energy of  $-112,153.63 \text{ [kJ/mol]}$  is calculated with mFES using  $\epsilon_{\text{protein}} = 4$ ,  $\epsilon_{\text{solv}} = 80$  with moderate resolution ( $h_s = 0.5 \text{ \AA}$ ). The CPU time needed to generate the molecular model takes nearly 6.5 h [Intel Xeon X5690 with 3.47 GHz] performing the calculation on single CPU core. The model involves 3 812 730 grid points corresponding to the same number of linear equations in first order. The calculation of a second order result needs to solve 30 501 685 linear equations, which is a factor of 8 larger as the first order equation system. Solving the linear equation system in second order with two iteration steps takes approximately 3 min. This application demonstrates that mFES is capable to deal with very large molecules.



**Figure 17.** Electrostatic potential map of lysozyme (placed on the molecular surface given by PyMOL)<sup>98</sup> for two values of the ionic strength  $I$  in the range from  $-10$  to  $10 \text{ [kT/e]}$  computed with mFES using  $h_s = 0.5 \text{ \AA}$  and APBS with a lattice constant of  $h = 0.25 \text{ \AA}$ . The images are generated with PyMOL.<sup>98</sup>



**Figure 18.** Electrostatic potential map for the adenovirus (PDB id 4CWU)<sup>47</sup> with 193k atoms in the range from  $-40$  to  $40 \text{ [kT/e]}$  computed with mFES with moderate resolution ( $h_s = 0.5 \text{ \AA}$ ) visualized with PyMOL.<sup>98</sup>

## CONCLUSIONS

To the best of our knowledge, mFES is the first FE method, which is competitive with well-established FD methods and sufficiently robust to calculate electrostatic properties for large proteins accurately. A major problem to establish a robust FE method for proteins is connected with the generation of a faithful protein surface by regular triangles, which are then used as basis for regular tetrahedrons covering the volume inside and outside of the molecule with well controlled point densities. FE approaches such as NETGEN<sup>42</sup> are generally used by engineers to optimize product design. This FE approach is adapted to our problem by developing the mFES program that calculates electrostatic solvation energies of molecular systems robustly and within a reasonable amount of CPU time. mFES combines LSMS,<sup>52</sup> which computes a high resolution molecular surface and NETGEN,<sup>42</sup> which simplifies and regularizes the molecular surface and meshes the inside and outside volume of the protein by optimizing the molecular model generation. Compared to

FD methods the number of linear equations to solve the *l*PBE is reduced by one to two orders magnitude. This allows to use also noniterative direct methods to solve the linear equations as for instance with MUMPS.<sup>57–59</sup> Solving the *l*PBE on the tetrahedral molecular model is fast compared to FD methods.

The generation of the molecular surface, the heart of matter of the mFES method, is described in detail and it is demonstrated that results can vary significantly if the molecular surface is not modeled in sufficient detail. FD methods can produce correct results if the underlying cubic grids are sufficiently fine grained. However, they reach a memory limit when representing the molecular surface with a regular simple cubic grid in high resolution to obtain very precise results. The FE method uses an irregular grid with grid points precisely set on the molecular surface and grid point densities, which are adjusted to the volume regimes that need to be covered. This allows to focus with high grid point densities on the critical parts of the molecular model, while it can also reduce grid point densities dramatically toward the asymptotic boundary rendering focusing procedures needed for FD methods superfluous. Thus, the program mFES can use a considerably smaller number of grid points (being identical to the number of linear equations to be solved) than FD methods and still reaches converged results with much higher accuracy than the ones obtained with FD methods.

Using three-dimensional molecular models opens a new way of molecular modelling. mFES produces interchangeable surface files in STL format which can be modified easily with current 3D animation software (Blender, Maya) or viewed by VMD<sup>70</sup> or PyMOL<sup>98</sup> using OpenDX files. We recommend using NETGEN<sup>42</sup> to generate appropriate grid models of molecular surfaces to visualize them and to map electrostatic potentials on them. mFES is able to reuse its own or manually modified molecular surface to generate the tetrahedral molecular model. Since FE methods are continuously advanced in industry, there will be further progress of FE methods, which will be useful for faster and even more precise electrostatic calculations with mFES. mFES is available free of charge at <http://agknapp.chemie.fu-berlin.de/mfes>.

## ASSOCIATED CONTENT

### Supporting Information

Optimization procedure to regularize tetrahedrons using NETGEN. Detailed view of triangulated molecular surfaces for proteins. Constants used to calculate reference values. Numerical values for electrostatic solvation energy for the Born ion model using ionic strength. CPU time ratios solving the *l*PBE for different proteins. This material is available free of charge via the Internet at <http://pubs.acs.org>.

## AUTHOR INFORMATION

### Corresponding Author

\*E-mail: knapp@chemie.fu-berlin.de.

### Notes

The authors declare no competing financial interest.

## ACKNOWLEDGMENTS

This work has been financially supported by the Volkswagen Stiftung and the German research society (DFG) within SFB 1078. We thank Milan Hodoscek for helpful discussions and Tolga Can for insights into surface generation and providing us with the source code of LSMS.

## ABBREVIATIONS

BIE, boundary integral equations; bpti, bovine pancreatic trypsin inhibitor; CcO, cytochrome c oxidase; dof, degrees of freedom; FD, finite difference; FE, finite element; H-distance, Hausdorff distance; *l*PBE, linearized Poisson–Boltzmann equation; MUMPS, multifrontal massively parallel sparse direct solver; PBE, Poisson–Boltzmann equation; SES, solvent excluded surface; SOR, successive over-relaxation; vdW, van der Waals

## REFERENCES

- (1) Warwicker, J.; Watson, H. C. Calculation of the Electric Potential in the Active Site Cleft due to A-Helix Dipoles. *J. Mol. Biol.* **1982**, *157*, 671–679.
- (2) Warshel, A.; Sharma, P. K.; Kato, M.; Parson, W. W. Modeling Electrostatic Effects in Proteins. *Biochim. Biophys. Acta* **2006**, *1764*, 1647–1676.
- (3) Baker, N. A.; Sept, D.; Joseph, S.; Holst, M. J.; McCammon, J. A. Electrostatics of Nanosystems: Application to Microtubules and the Ribosome. *Proc. Natl. Acad. Sci. U.S.A.* **2001**, *98*, 10037–10041.
- (4) Nicholls, A.; Honig, B. A Rapid Finite Difference Algorithm, Utilizing Successive Over-Relaxation to Solve the Poisson–Boltzmann Equation. *J. Comput. Chem.* **1991**, *12*, 435–445.
- (5) Bashford, D. An Object-Oriented Programming Suite for Electrostatic Effects in Biological Molecules. In *Proceedings of the first International Conference, ISCOPE*; Ishikawa, Y., Oldehoeft, R. R., Ed.; Springer: Marina del Rey, CA, 1997; Vol. 1343, pp 233–240.
- (6) Antosiewicz, J.; Briggs, J. M.; Elcock, A. H.; Gilson, M. K.; McCammon, J. A. Computing Ionization States of Proteins with a Detailed Charge Model. *J. Comput. Chem.* **1996**, *17*, 1633–1644.
- (7) Boschitsch, A. H.; Fenley, M. O. A Fast and Robust Poisson–Boltzmann Solver Based on Adaptive Cartesian Grids. *J. Chem. Theory Comput.* **2011**, *7*, 1524–1540.
- (8) Gilson, M. K.; Sharp, K. A.; Honig, B. H. Calculating the Electrostatic Potential of Molecules in Solution: Method and Error Assessment. *J. Comput. Chem.* **1988**, *9*, 327–335.
- (9) *Biomolecular Applications of Poisson–Boltzmann Methods*; Reviews in Computational Chemistry; Lipkowitz, K. B.; Larter, R.; Cundari, T. R., Eds.; John Wiley & Sons, Inc.: Hoboken, NJ, 2005.
- (10) Mohammad Mirzadeh, M. T. An Adaptive, Finite Difference Solver for the Nonlinear Poisson–Boltzmann Equation with Applications to Biomolecular Computations. *Commun. Comput. Phys.* **2012**, *13*, 150–173.
- (11) Lu, B.; Zhang, D.; McCammon, J. A. Computation of Electrostatic Forces between Solvated Molecules Determined by the Poisson–Boltzmann Equation Using a Boundary Element Method. *J. Chem. Phys.* **2005**, *122*, 214102.
- (12) Zauhar, R. J.; Morgan, R. S. A New Method for Computing the Macromolecular Electric Potential. *J. Mol. Biol.* **1985**, *186*, 815–820.
- (13) Zauhar, R. J.; Morgan, R. S. The Rigorous Computation of the Molecular Electric Potential. *J. Comput. Chem.* **1988**, *9*, 171–187.
- (14) Boschitsch, A. H.; Fenley, M. O.; Zhou, H.-X. Fast Boundary Element Method for the Linear Poisson–Boltzmann Equation. *J. Phys. Chem. B* **2002**, *106*, 2741–2754.
- (15) Altman, M. D.; Bardhan, J. P.; Tidor, B.; White, J. K. FFTSVD: A Fast Multiscale Boundary-Element Method Solver Suitable for Bio-MEMS and Biomolecule Simulation. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2006**, *25*, 274–284.
- (16) Lu, B.; McCammon, J. A. Improved Boundary Element Methods for Poisson–Boltzmann Electrostatic Potential and Force Calculations. *J. Chem. Theory Comput.* **2007**, *3*, 1134–1142.
- (17) Liang, J.; Subramaniam, S. Computation of Molecular Electrostatics with Boundary Element Methods. *Biophys. J.* **1997**, *73*, 1830–1841.
- (18) Zauhar, R. J.; Varnek, A. A Fast and Space-Efficient Boundary Element Method for Computing Electrostatic and Hydration Effects in Large Molecules. *J. Comput. Chem.* **1996**, *17*, 864–877.
- (19) Lu, B. Z.; Zhou, Y. C.; Holst, M. J.; McCammon, J. A. Recent Progress in Numerical Methods for the Poisson–Boltzmann Equation

- in Biophysical Applications. *Commun. Comput. Phys.* **2008**, *3*, 973–1009.
- (20) Lu, B.; Cheng, X.; Hou, T.; McCammon, J. A. Calculation of the Maxwell Stress Tensor and the Poisson–Boltzmann Force on a Solvated Molecular Surface Using Hypersingular Boundary Integrals. *J. Chem. Phys.* **2005**, *123*, 084904.
- (21) Yang, S. A. On the Singularities of Green's Formula and Its Normal Derivative, with an Application to Surface-Wave-Body Interaction Problems. *Int. J. Numer. Methods Eng.* **2000**, *47*, 1841–1864.
- (22) Bochevarov, A. D.; Harder, E.; Hughes, T. F.; Greenwood, J. R.; Braden, D. A.; Philipp, D. M.; Rinaldo, D.; Halls, M. D.; Zhang, J.; Friesner, R. A. Jaguar: A High-Performance Quantum Chemistry Software Program with Strengths in Life and Materials Sciences. *Int. J. Quantum Chem.* **2013**, *113*, 2110–2142.
- (23) Cortis, C. M.; Friesner, R. A. An Automatic Three-Dimensional Finite Element Mesh Generation System for the Poisson–Boltzmann Equation. *J. Comput. Chem.* **1997**, *18*, 1570–1590.
- (24) Cortis, C. M.; Friesner, R. A. Numerical Solution of the Poisson–Boltzmann Equation Using Tetrahedral Finite-Element Meshes. *J. Comput. Chem.* **1997**, *18*, 1591–1608.
- (25) Li, L.; Li, C.; Sarkar, S.; Zhang, J.; Witham, S.; Zhang, Z.; Wang, L.; Smith, N.; Petukh, M.; Alexov, E. DelPhi: A Comprehensive Suite for DelPhi Software and Associated Resources. *BMC Biophys.* **2012**, *5*, 9.
- (26) Holst, M.; McCammon, J. A.; Yu, Z.; Zhou, Y.; Zhu, Y. Adaptive Finite Element Modeling Techniques for the Poisson–Boltzmann Equation. *Commun. Comput. Phys.* **2012**, *11*, 179–214.
- (27) Holst, M.; Saied, F. Multigrid Solution of the Poisson–Boltzmann Equation. *J. Comput. Chem.* **1993**, *14*, 105–113.
- (28) Holst, M.; Szypowski, R.; Zhu, Y. Adaptive Finite Element Methods with Inexact Solvers for the Nonlinear Poisson–Boltzmann Equation. *Domain Decompos. Methods Sci. Eng. XX* **2013**, *91*, 167–174.
- (29) Chen, L.; Holst, M. J.; Xu, J. The Finite Element Approximation of the Nonlinear Poisson–Boltzmann Equation. *SIAM J. Numer. Anal.* **2007**, *45*, 2298–2320.
- (30) Decherchi, S.; Colmenares, J.; Catalano, C. E.; Spagnuolo, M.; Alexov, E.; Rocchia, W. Between Algorithm and Model: Different Molecular Surface Definitions for the Poisson–Boltzmann Based Electrostatic Characterization of Biomolecules in Solution. *Commun. Comput. Phys.* **2013**, *13*, 61–89.
- (31) Grant, J. A.; Pickup, B. T. A Gaussian Description of Molecular Shape. *J. Phys. Chem.* **1995**, *99*, 3503–3510.
- (32) Swanson, J. M. J.; Mongan, J.; McCammon, J. A. Limitations of Atom-Centered Dielectric Functions in Implicit Solvent Models. *J. Phys. Chem. B* **2005**, *109*, 14769–14772.
- (33) Holst, M. Adaptive Numerical Treatment of Elliptic Systems on Manifolds. *Adv. Comput. Math.* **2001**, *15*, 139–191.
- (34) Si, H.Gärtner, K. Meshing Piecewise Linear Complexes by Constrained Delaunay Tetrahedralizations. In *Proceedings of the 14th International Meshing Roundtable*; Bryan, W. H., Ed.; Springer: Berlin, Germany, 2005; pp 147–163.
- (35) Born, M. Volumen Und Hydratationswärme Der Ionen. *Z. Phys.* **1920**, *45*–48.
- (36) Wan, Z.; Xu, B.; Huang, K.; Chu, Y.-C.; Li, B.; Nakagawa, S. H.; Qu, Y.; Hu, S.-Q.; Katsoyannis, P. G.; Weiss, M. A. Enhancing the Activity of Insulin at the Receptor Interface: Crystal Structure and Photo-Cross-Linking of A8 Analogues. *Biochemistry* **2004**, *43*, 16119–16133.
- (37) APBS Software; <http://www.poissonboltzmann.org/apbs/> (accessed Apr. 8, 2014).
- (38) Bond, S. D.; Chaudhry, J. H.; Cyr, E. C.; Olson, L. N. A First-Order System Least-Squares Finite Element Method for the Poisson–Boltzmann Equation. *J. Comput. Chem.* **2010**, *31*, 1625–1635.
- (39) Wenbin, C.; Yifan, S.; Qing, X. A Mortar Finite Element Approximation for the Linear Poisson–Boltzmann Equation. *Appl. Math. Comput.* **2005**, *164*, 11–23.
- (40) Belgacem, F. B.; Hild, P.; Laborde, P. The Mortar Finite Element Method for Contact Problems. *Math. Comput. Model.* **1998**, *28*, 263–271.
- (41) Xie, D.; Zhou, S. A New Minimization Protocol for Solving Nonlinear Poisson–Boltzmann Mortar Finite Element Equation. *BIT Numer. Math.* **2007**, *47*, 853–871.
- (42) Schöberl, J. NETGEN An Advancing Front 2D/3D-Mesh Generator Based on Abstract Rules. *Comput. Vis. Sci.* **1997**, *1*, 41–52.
- (43) Berndt, K. D.; Güntert, P.; Orbons, L. P.; Wüthrich, K. Determination of a High-Quality Nuclear Magnetic Resonance Solution Structure of the Bovine Pancreatic Trypsin Inhibitor and Comparison with Three Crystal Structures. *J. Mol. Biol.* **1992**, *227*, 757–775.
- (44) Martin, C.; Richard, V.; Salem, M.; Hartley, R.; Mauguin, Y. Refinement and Structural Analysis of Barnase at 1.5 Å Resolution. *Acta Crystallogr. D. Biol. Crystallogr.* **1999**, *55*, 386–398.
- (45) Ramanadham, M.; Sieker, L. C.; Jensen, L. H. Refinement of Triclinic Lysozyme: II. The Method of Stereochemically Restrained Least Squares. *Acta Crystallogr. B* **1990**, *46* (Pt 1), 63–69.
- (46) Svensson-Ek, M.; Abramson, J.; Larsson, G.; Törnroth, S.; Brzezinski, P.; Iwata, S. The X-Ray Crystal Structures of Wild-Type and EQ(I-286) Mutant Cytochrome c Oxidases from *Rhodobacter sphaeroides*. *J. Mol. Biol.* **2002**, *321*, 329–339.
- (47) Reddy, V. S.; Nemerow, G. R. Structures and Organization of Adenovirus Cement Proteins Provide Insights into the Role of Capsid Maturation in Virus Entry and Infection. *Proc. Natl. Acad. Sci. U.S.A.* **2014**, *111*, 11715–11720.
- (48) Takashima, S.; Schwan, H. P. Dielectric Dispersion of Crystalline Powders of Amino Acids, Peptides, and Proteins I. *J. Phys. Chem.* **1965**, *69*, 4176–4182.
- (49) Schutz, C. N.; Warshel, A. What Are the Dielectric “Constants” of Proteins and How to Validate Electrostatic Models? *Proteins* **2001**, *44*, 400–417.
- (50) Kieseritzky, G.; Knapp, E.-W. Optimizing pK<sub>a</sub> Computation in Proteins with pH Adapted Conformations. *Proteins* **2008**, *71*, 1335–1348.
- (51) Kieseritzky, G. Shaping Electrostatic Energy Computations in Proteins: The ClC-Type Proton-Chloride Antiporter Function. *Dissertation* **2011**, 1–64.
- (52) Can, T.; Chen, C.-I.; Wang, Y.-F. Efficient Molecular Surface Generation Using Level-Set Methods. *J. Mol. Graph. Modell.* **2006**, *25*, 442–454.
- (53) Zhou, Z.; Payne, P.; Vasquez, M.; Kuhn, N.; Levitt, M. Finite-Difference Solution of the Poisson–Boltzmann Equation: Complete Elimination of Self-Energy. *J. Comput. Chem.* **1996**, *17*, 1344–1351.
- (54) Abramowitz, M.; Stegun, I. A. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*; Dover Publications: New York, 1965; p 1046.
- (55) Polycarpou, A. C. *Introduction to the Finite Element Method in Electromagnetics; Synthesis*; Morgan & Claypool Publishers: San Rafael, CA, 2006; p 120.
- (56) Haji-Akbari, A.; Engel, M.; Keys, A. S.; Zheng, X.; Petschek, R. G.; Palffy-Muhoray, P.; Glotzer, S. C. Disordered, Quasicrystalline, and Crystalline Phases of Densely Packed Tetrahedra. *Nature* **2009**, *462*, 773–777.
- (57) Amestoy, P. R.; Davis, T. A.; Duff, I. S. An Approximate Minimum Degree Ordering Algorithm. *SIAM J. Matrix Anal. Appl.* **1996**, *17*, 886–905.
- (58) Amestoy, P. R.; Duff, I.; L'Excellent, J. Y. *MUMPS Multifrontal Massively Parallel Solver*, Tech. Report, TR/PA/98/02; 1998.
- (59) Amestoy, P. R.; Duff, I. S.; L'Excellent, J.-Y. Multifrontal Parallel Distributed Symmetric and Unsymmetric Solvers. *Comput. Methods Appl. Mech. Eng.* **2000**, *184*, 501–520.
- (60) Lee, B.; Richards, F. M. The Interpretation of Protein Structures: Estimation of Static Accessibility. *J. Mol. Biol.* **1971**, *55*, 379–400.
- (61) Richards, F. M. Areas, Volumes, Packing, and Protein Structure. *Annu. Rev. Biophys. Bioeng.* **1977**, *6*, 151–176.
- (62) Greer, J.; Bush, B. L. Macromolecular Shape and Surface Maps by Solvent Exclusion. *Proc. Natl. Acad. Sci. U.S.A.* **1978**, *75*, 303–307.
- (63) Sanner, M. F.; Olson, A. J.; Spehner, J. C. Reduced Surface: An Efficient Way to Compute Molecular Surfaces. *Biopolymers* **1996**, *38*, 305–320.

- (64) Connolly, M. L. Analytical Molecular Surface Calculation. *J. Appl. Crystallogr.* **1983**, *16*, 548–558.
- (65) Ullmann, G. M.; Knapp, E. W. Electrostatic Models for Computing Protonation and Redox Equilibria in Proteins. *Eur. Biophys. J.* **1999**, *28*, 533–551.
- (66) Xu, D.; Zhang, Y. Generating Triangulated Macromolecular Surfaces by Euclidean Distance Transform. *PLoS One* **2009**, *4*, e8140.
- (67) Bernardini, F.; Mittleman, J.; Rushmeier, H.; Silva, C.; Taubin, G. The Ball-Pivoting Algorithm for Surface Reconstruction. *IEEE Trans. Vis. Comput. Graph* **1999**, *5*, 349–359.
- (68) Chen, M.; Lu, B. Advances in Biomolecular Surface Meshing and Its Applications to Mathematical Modeling. *Chin. Sci. Bull.* **2013**, *58*, 1843–1849.
- (69) Tomasi, J.; Mennucci, B.; Cammi, R. Quantum Mechanical Continuum Solvation Models. *Chem. Rev.* **2005**, *105*, 2999–3093.
- (70) Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual Molecular Dynamics. *J. Mol. Graph* **1996**, *14* (33–38), 27–28.
- (71) Goddard, T. D.; Huang, C. C.; Ferrin, T. E. Software Extensions to UCSF Chimera for Interactive Visualization of Large Molecular Assemblies. *Structure* **2005**, *13*, 473–482.
- (72) Connolly, M. L. Molecular Surface Triangulation. *J. Appl. Crystallogr.* **1985**, *18*, 499–505.
- (73) Lorensen, W. E.; Cline, H. E. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *ACM SIGGRAPH Comput. Graph* **1987**, *21*, 163–169.
- (74) Decherchi, S.; Rocchia, W. A General and Robust Ray-Casting-Based Algorithm for Triangulating Surfaces at the Nanoscale. *PLoS One* **2013**, *8*, e59744.
- (75) Cgal: Computational Geometry Algorithms Library <http://www.cgal.org>.
- (76) Osher, S.; Fedkiw, R.; Piechor, K. Level Set Methods and Dynamic Implicit Surfaces. *Appl. Mech. Rev.* **2002**, *57*, B15.
- (77) Guex, N.; Peitsch, M. C. SWISS-MODEL and the Swiss-PdbViewer: An Environment for Comparative Protein Modeling. *Electrophoresis* **1997**, *18*, 2714–2723.
- (78) Taubin, G. Curve and Surface Smoothing without Shrinkage. In *Proceedings of 5th IEEE International Conference on Computer Vision*; IEEE Comput. Soc. Press: Cambridge, MA, 1995; pp 852–857.
- (79) Davies, R. *Computer and Machine Vision: Theory, Algorithms, Practicalities*; Academic Press: Waltham, MA, 1990.
- (80) Jin, H.; Tanner, R. I. Generation of Unstructured Tetrahedral Meshes by Advancing Front Technique. *Int. J. Numer. Methods Eng.* **1993**, *36*, 1805–1823.
- (81) George, P. L.; Sevano, E. The Advancing-Front Mesh Generation Method Revisited. *Int. J. Numer. Methods Eng.* **1994**, *37*, 3605–3619.
- (82) Pang, X.; Zhou, H.-X. Poisson–Boltzmann Calculations: Van Der Waals or Molecular Surface? *Commun. Comput. Phys.* **2013**, *13*, 1–12.
- (83) Hoppe, H.; DeRose, T.; Duchamp, T.; McDonald, J.; Stuetzle, W. Mesh Optimization. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques—SIGGRAPH '93*; Whitton, M. C., Cameron, G., Ed.; ACM Press: New York, 1993; pp 19–26.
- (84) Lori, A.; Freitag, C. O. Tetrahedral Mesh Improvement Using Swapping and Smoothing. *Int. J. Numer. Methods Eng.* **1997**, *40*, 3979–4002.
- (85) Kirkwood, J. G. Theory of Solutions of Molecules Containing Widely Separated Charges with Special Application to Zwitterions. *J. Chem. Phys.* **1934**, *2*, 351.
- (86) Holst, M. J. The Poisson–Boltzmann Equation: Analysis and Multilevel Numerical Solution. PhD thesis, Numerical Computing Group, University of Illinois at Urbana-Champaign, Champaign, IL, Oct. 1993.
- (87) Mohr, P. J.; Taylor, B. N.; Newell, D. B. CODATA Recommended Values of the Fundamental Physical Constants: 2010. *J. Phys. Chem. Ref. Data* **2012**, *41*, 043109.
- (88) Berger, M. J.; Colella, P. Local Adaptive Mesh Refinement for Shock Hydrodynamics. *J. Comput. Phys.* **1989**, *82*, 64–84.
- (89) Debye, P.; Hückel, E. Zur Theorie Der Elektrolyte. I. Gefrierpunktserniedrigung Und Verwandte Erscheinungen. The Theory of Electrolytes. I. Lowering of Freezing Point and Related Phenomena. *Phys. Z.* **1923**, *24*, 185–206.
- (90) Brooks, B. R.; Brucoleri, R. E.; Olafson, B. D.; Statese, D. J.; Swaminathan, S.; Karplus, M. CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations. *J. Comput. Chem.* **1983**, *4*, 187–217.
- (91) Alexander, D. MacKerell, B. B. CHARMM: The Energy Function and Its Parameterization with an Overview of the Program. In *Encyclopedia of Computational Chemistry*; von Ragué Schleyer, P.; Allinger, N. L.; Clark, T.; Gasteiger, J.; Kollman, P. A.; Schaefer, H. F.; Schreiner, P. R., Eds.; John Wiley & Sons, Ltd: Chichester, U.K., 2002; pp 271–277.
- (92) Brooks, B. R.; Brooks, C. L.; Mackerell, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; Caflisch, A.; Caves, L.; Cui, Q.; Dinner, A. R.; Feig, M.; Fischer, S.; Gao, J.; Hodoscek, M.; Im, W.; Kuczera, K.; Lazaridis, T.; Ma, J.; Ovchinnikov, V.; Paci, E.; Pastor, R. W.; Post, C. B.; Pu, J. Z.; Schaefer, M.; Tidor, B.; Venable, R. M.; Woodcock, H. L.; Wu, X.; Yang, W.; York, D. M.; Karplus, M. CHARMM: The Biomolecular Simulation Program. *J. Comput. Chem.* **2009**, *30*, 1545–1614.
- (93) Knapp, E. W.; Muegge, I. Heterogeneous Diffusion of Water at Protein Surfaces: Application to BPTI. *J. Phys. Chem.* **1993**, *97*, 11339–11343.
- (94) Woelke, A. L.; Galstyan, G.; Galstyan, A.; Meyer, T.; Heberle, J.; Knapp, E.-W. Exploring the Possible Role of Glu286 in CCO by Electrostatic Energy Computations Combined with Molecular Dynamics. *J. Phys. Chem. B* **2013**, *117*, 12432–12441.
- (95) Huttenlocher, D. P.; Klanderman, G. A.; Rucklidge, W. J. Comparing Images Using the Hausdorff Distance. *IEEE Trans. Pattern Anal. Mach. Intell.* **1993**, *15*, 850–863.
- (96) Cignoni, P.; Rocchini, C.; Scopigno, R. Metro: Measuring Error on Simplified Surfaces. *Comput. Graph. Forum* **1998**, *17*, 167–174.
- (97) Cignoni, P.; Corsini, M.; Ranzuglia, G. *MeshLab: An Open-Source 3D Mesh Processing System*; ISTI-CNR : Italy.
- (98) Schrödinger, L. *The PyMOL Molecular Graphics System*; Schrödinger, LLC: New York.