# Supporting Information

# Electrostatic and proton–electron coupling in membrane-bound NrfH under external electric fields: A computational study

Zeinab Rahimi[1], Amir Lohrasebi[2] *, Thorsten Koslowski[1]

[1]Institute for Physical Chemistry, University of Freiburg, Albertstrasse 21, D-79104 Freiburg, Germany

[2]Department of Physics, University of Isfahan, Isfahan, 8174673441, Iran

*Corresponding author, e-mail: lohrasebi@nano.ipm.ac.ir

## 1. Electrostatic Calculations and Transmembrane Potential Mapping

In our model, it is located in a heterogeneous polarizable environment, characterized by the dielectric constant $\varepsilon(r)$ of the surrounding water, the membrane and the protein itself. Taking the protein partial charges $q_i$ into account, it can be described by the Poisson equation,

$$\nabla.\varepsilon(r)\nabla\varphi(r) = \rho(r) = \sum_i q_i$$

The numerical solution of this equation (and the related Poisson-Boltzmann equation) is based on the projection of the protein and its environment onto a lattice and the consecutive solution of the emerging sparse system of equations for the potential $\varphi$. This task has been implemented in program packages like Delphi, APBS or H++, a list by no means completes [1-5]. To the best of our knowledge, none of these packages permits the application of an external potential via boundary conditions. As charges average out on a sufficiently large length scale due to the requirement of local charge neutrality, and as our main focus is on the general behaviour of the potential along the system, we have made use of the corresponding Laplace equation:

$$\nabla.\varepsilon(r)\nabla\varphi(r) = 0$$

It retains the dielectric heterogeneity, which reflects the subdivision of the system into protein, membrane, and the aqueous environment.

We write the discretized Nabla term as a Laplacian molecule, discretizing the partial derivatives on a cubic lattice with a spacing h and with indices i, j, k for the sites, we have:

$$\nabla\varepsilon \simeq \left(\varepsilon_{i+1,j,k} + \varepsilon_{i,j+1,k} + \varepsilon_{i,j,k+1} - 3\varepsilon_{i,j,k}\right)/h$$

a similar expression for $\nabla\varphi$ and

$$\nabla^2\varphi \simeq \frac{\left(\varphi_{i+1,j,k} + \varphi_{i,j+1,k} + \varphi_{i,j,k+1} - 6\varphi_{i,j,k} + \varphi_{i-1,j,k} + \varphi_{i,j-1,k} + \varphi_{i,j,k-1}\right)}{6h^2}$$

These combine into:

$$\{\varepsilon_{i+1,j,k}\varphi_{i+1,j,k} + \varepsilon_{i,j+1,k}\varphi_{i,j+1,k} + \varepsilon_{i,j,k+1}\varphi_{i,j,k+1} - (3\varepsilon_{i,j,k} + \varepsilon_{i+1,j,k} + \varepsilon_{i,j+1,k} +$$
$$= 0$$

and an iteration scheme,

$$\varphi_{i,j,k}^{(new)} = \frac{\left\{\begin{matrix}\varepsilon_{i,j,k}\left(\varphi_{i+1,j,k}^{(old)} + \varphi_{i,j+1,k}^{(old)} + \varphi_{i,j,k+1}^{(old)}\right) + \varepsilon_{i+1,j,k}\varphi_{i+1,j,k}^{(old)} \\ + \varepsilon_{i,j+1,k}\varphi_{i,j+1,k}^{(old)} + \varepsilon_{i,j,k+1}\varphi_{i,j+1,k}^{(old)}\end{matrix}\right\}}{\left(3\varepsilon_{i,j,k} + \varepsilon_{i+1,j,k} + \varepsilon_{i,j+1,k} + \varepsilon_{i,j,k+1}\right)}$$

Boundary conditions in the z direction correspond to fixed potentials at z=0 and at $z_{\max}$, the maximum extension of the system in that direction. Along the x and y directions, the minimum image convention is used, thus reducing the possible influence of surface effects. As the Laplace equation is a linear partial differential equation, the potentials $\varphi$ scale linearly with the applied external potential, and a single computation suffices for each system geometry.

Numerically, the discretized Laplace equations is solved by the successive overrelaxation (SOR) method, motivated by the approach used in the Delphi program package. Here, we switch between two types of sites on a bipartite lattice for each iteration. Typically, we use a grid of 64 sites along the z direction and at least as many sites into the x and y directions. Usually, a few hundred iterations are required to achieve relative accuracy in $\Delta\varphi/\varphi$ of $10^{-10}$. To project the potential back onto the positions of the protein atoms, a trilinear interpolation scheme is used.

## References

1. Jurrus E, Engel D, Star K, Monson K, Brandi J, Felberg LE et al (2018) Improvements to the APBS biomolecular solvation software suite. Protein Sci 27:112–128

2. Konecny R, Baker NA, McCammon JA (2012) iAPBS: a programming interface to Adaptive Poisson-Boltzmann Solver (APBS). Comput Sci Discov 5:015005

3. Li L, Li C, Sarkar S, Zhang J, Witham S, Zhang Z et al (2012) DelPhi: a comprehensive suite for DelPhi software and associated resources. BMC Biophys 5:9

4. Li C, Jia Z, Chakravorty A, Pahari S, Peng Y, Basu S et al (2019) DelPhi Suite: New Developments and Review of Functionalities. J Comput Chem 40:2502–2508

5. Anandakrishnan G, Aguilar B, Onufriev AV (2012) H++ 3.0: automating pK prediction and the preparation of biomolecular structures for atomistic molecular modeling and simulations. Nucleic Acids Res 40:W537–W541