MATLAB version number 5 of the APBS

(Dated: November 30, 2009)

Abstract

This solver uses the biconjugate gradient stabilized method and the inexact LU decomposition to numerically solve the linearized PB equation on a cartesian 3D-grid. This version requires the shifted dielectric and the ion accessibility coefficient (kappa function) maps as generated by the APBS code as well as the corresponding pqr file generated by the pdb2pqr code. It uses standard three-linear splines (spl0) to spread the charge density along the nearest grid points if needed. It is able to solve the linear PB eq with either Dirichlet or focus boundary conditions. In the later case, this code solves the PB equation in a large (low resolution) domain and the resulting electrostatic potential solution is subsequently used to evaluate the Dirichlet boundary condition to solve the PB equation in a (higher resolution) sub-domain region. The resulting electrostatic potential and charge maps for both the coarse and target grids are saved in dx format in different folders. For visualization purpose, this code also generates two files (.fig and .tiff) corresponding to the graphical representation of the electrostatic potential surface. This version was used to solve the pka example provided by the apbs package. If the Dirichlet Boundary Condition is required, then this version basically provides the same results than the previous one. The main difference is that the user doesn't have to edit the source files in this version but only have to provide the target input-file name and the corresponding full path as the only argument of the (new) matlab function MAPBS (x). See the pka example and the MATLAB_PB_SOLVER_5 package for more details.

Description

This code is based on Michel Holst's thesis and Nathan Baker's APBS approach. The box-method is used to discretize the following (linearized) PB equation

$$-\nabla.\left(\epsilon\left(\mathbf{r}\right)\nabla u\left(\mathbf{r}\right)\right)+\bar{\kappa}\left(\mathbf{r}\right)^{2}u\left(\mathbf{r}\right)=magic\sum_{i=1}^{N}z_{i}\delta\left(\mathbf{r}-\mathbf{r}_{i}\right) \tag{1}$$

where $u\left(\mathbf{r}\right)=e_{c}\Phi\left(\mathbf{r}\right)/K_{B}T$ and $magic=4\pi e_{c}^{2}/K_{B}T$. For a diagonal dielectric tensor, the resulting discretized linear PB equations at the nodes $u_{ijk}=u\left(x_{i},y_{j},z_{k}\right)$ for $1\leq i\leq N_{x}$, $1\leq j\leq N_{y}$ and $1\leq k\leq N_{z}$ reads

$$\left[\epsilon_{i-1/2,j,k}^{x}\frac{\left(h_{j-1}+h_{j}\right)\left(h_{k-1}+h_{k}\right)}{4h_{i-1}}+\epsilon_{i+1/2,j,k}^{x}\frac{\left(h_{j-1}+h_{j}\right)\left(h_{k-1}+h_{k}\right)}{4h_{i}}+\right.$$

$$\epsilon_{i,j-1/2,k}^{y}\frac{\left(h_{i-1}+h_{i}\right)\left(h_{k-1}+h_{k}\right)}{4h_{j-1}}+\epsilon_{i,j+1/2,k}^{y}\frac{\left(h_{i-1}+h_{i}\right)\left(h_{k-1}+h_{k}\right)}{4h_{j}}+$$

$$\epsilon_{i,j,k-1/2}^{k}\frac{\left(h_{i-1}+h_{i}\right)\left(h_{j-1}+h_{j}\right)}{4h_{k-1}}+\epsilon_{i,j,k+1/2}^{k}\frac{\left(h_{i-1}+h_{i}\right)\left(h_{j-1}+h_{j}\right)}{4h_{k}}+$$

$$\left.\kappa_{ijk}\frac{\left(h_{i-1}+h_{i}\right)\left(h_{j-1}+h_{j}\right)\left(h_{k-1}+h_{k}\right)}{8}\right]u_{ijk}+$$

$$\left[-\epsilon_{i-1/2,j,k}^{x}\frac{\left(h_{j-1}+h_{j}\right)\left(h_{k-1}+h_{k}\right)}{4h_{i-1}}\right]u_{i-1jk}+\left[-\epsilon_{i+1/2,j,k}^{x}\frac{\left(h_{j-1}+h_{j}\right)\left(h_{k-1}+h_{k}\right)}{4h_{i}}\right]u_{i+1jk}+$$

$$\left[-\epsilon_{i,j-1/2,k}^{y}\frac{\left(h_{i-1}+h_{i}\right)\left(h_{k-1}+h_{k}\right)}{4h_{j-1}}\right]u_{ij-1k}+\left[-\epsilon_{i,j+1/2,k}^{y}\frac{\left(h_{i-1}+h_{i}\right)\left(h_{k-1}+h_{k}\right)}{4h_{j}}\right]u_{ij+1k}+$$

$$\left[-\epsilon_{i,j,k-1/2}^{k}\frac{\left(h_{i-1}+h_{i}\right)\left(h_{j-1}+h_{j}\right)}{4h_{k-1}}\right]u_{ijk-1}+\left[-\epsilon_{i,j,k+1/2}^{k}\frac{\left(h_{i-1}+h_{i}\right)\left(h_{j-1}+h_{j}\right)}{4h_{k}}\right]u_{ijk+1}=$$

$$magic\frac{\left(h_{i-1}+h_{i}\right)\left(h_{j-1}+h_{j}\right)\left(h_{k-1}+h_{k}\right)}{8}f_{ijk} \tag{2}$$

in which

$$h_{i}=x_{i+1}-x_{i},\ h_{j}=y_{j+1}-y_{j}\ h_{k}=z_{k+1}-z_{k}$$

The delta functions appearing in the right hand side of the starting equations are approximated with linear B-splines (spl0) which spread the point like charge along the nearest neighborhood. The resulting f_{ijk} represent the smearing of the point charges along the grid points.

For more details, including used unit system, please refer to the Michel Holst's thesis and the APBS user guide online. To visualize more clearly the problem, let's explicitly write the first equations for a cubic grid of 5x5x5 containing general coefficients

$$a_{222}u_{222} + a_{122}u_{122} + a_{322}u_{322} + a_{212}u_{212} + a_{232}u_{232} + a_{221}u_{221} + a_{223}u_{223} = f_{222}$$

$$a_{322}u_{322} + a_{222}u_{222} + a_{422}u_{422} + a_{312}u_{312} + a_{332}u_{332} + a_{321}u_{321} + a_{323}u_{323} = f_{322}$$

$$a_{422}u_{422} + a_{122}u_{122} + a_{322}u_{322} + a_{212}u_{212} + a_{232}u_{232} + a_{221}u_{221} + a_{223}u_{223} = f_{422}$$

$$a_{232}u_{232} + a_{132}u_{132} + a_{332}u_{332} + a_{222}u_{222} + a_{242}u_{242} + a_{231}u_{231} + a_{233}u_{233} = f_{232}$$

$$\dots$$

in which the nodes are arranged using the natural ordering

$$U = [u_{111}, u_{211}, .., u_{N_x11}, u_{121}, .., u_{221}, u_{321}, .., u_{N_x21} ..., u_{N_xN_yN_z}]^T$$

Note that the prescribed values of nodes $u_{1jk}, u_{N_x,j,k}, u_{i,1,k}, u_{i,N_y,k}, u_{ij1}$ and u_{ijN_z} along the faces of the box coming from the Dirichlet boundary conditions will have their corresponding elements removed in such a way that only equations for the interior nodes remain. In other words, we will only consider the following set of unknown nodes

$$U = [u_{222}, u_{322}, .., u_{N_x-1,22}, u_{232}, .., u_{332}, u_{432}, .., u_{N_x-2,32} ..., u_{N_x-1,N_y-1,N_z-1}]^T$$

in such a way that the previous equations become

$$a_{222}u_{222} + a_{322}u_{322} + a_{232}u_{232} + a_{223}u_{223} = f_{222} - a_{122}u_{122} - a_{212}u_{212} - a_{221}u_{221} \equiv b_{222}$$

$$a_{322}u_{322} + a_{222}u_{222} + a_{422}u_{422} + a_{332}u_{332} + a_{323}u_{323} = f_{322} - a_{312}u_{312} - a_{321}u_{321} \equiv b_{322}$$

$$a_{422}u_{422} + a_{322}u_{322} + a_{232}u_{232} + a_{223}u_{223} = f_{422} - a_{122}u_{122} - a_{212}u_{212} - a_{221}u_{221} \equiv b_{422}$$

$$a_{232}u_{232} + a_{332}u_{332} + a_{222}u_{222} + a_{242}u_{242} + a_{233}u_{233} = f_{232} - a_{132}u_{132} - a_{231}u_{231} \equiv b_{232}$$

in which the boundary $u's$ are conveniently brought to the right-hand-side of the equations. The resulting left-hand side equations can be written in compact form in term of matrix vector product as follows

$$Au = b$$

in which

$$u\left(p\right) = u_{ijk}, \qquad b\left(p\right) = b_{ijk}, \qquad p = (k-2)(N_x-2)(N_y-2) + (j-2)(N_x-2) + i - 1$$

$$i = 2, .., N_x - 2, \quad j = 2, .., N_y - 2, \quad k = 2, .., N_z - 2$$

and A is a (seven banded block tri-diagonal form) $(N_x-2)(N_y-2)(N_z-2)$ by $(N_x-2)(N_y-2)(N_z-2)$ squared symmetric positive definite matrix containing the following nonzero elements (see figure 1):

- The main diagonal elements

$$d_0(p) = \left[\epsilon^x_{i-1/2,j,k}\frac{\left(h_{j-1}+h_j\right)\left(h_{k-1}+h_k\right)}{4h_{i-1}} + \epsilon^x_{i+1/2,j,k}\frac{\left(h_{j-1}+h_j\right)\left(h_{k-1}+h_k\right)}{4h_i} + \right.$$

$$\epsilon^y_{i,j-1/2,k}\frac{\left(h_{i-1}+h_i\right)\left(h_{k-1}+h_k\right)}{4h_{j-1}} + \epsilon^y_{i,j+1/2,k}\frac{\left(h_{i-1}+h_i\right)\left(h_{k-1}+h_k\right)}{4h_j} +$$

$$\epsilon^k_{i,j,k-1/2}\frac{\left(h_{i-1}+h_i\right)\left(h_{j-1}+h_j\right)}{4h_{k-1}} + \epsilon^k_{i,j,k+1/2}\frac{\left(h_{i-1}+h_i\right)\left(h_{j-1}+h_j\right)}{4h_k} +$$

$$\left.\kappa_{ijk}\frac{\left(h_{i-1}+h_i\right)\left(h_{j-1}+h_j\right)\left(h_{k-1}+h_k\right)}{8}\right] \tag{3}$$

- The Next upper band diagonal, which is shifted in one column to the left from the first column, contains the following elements

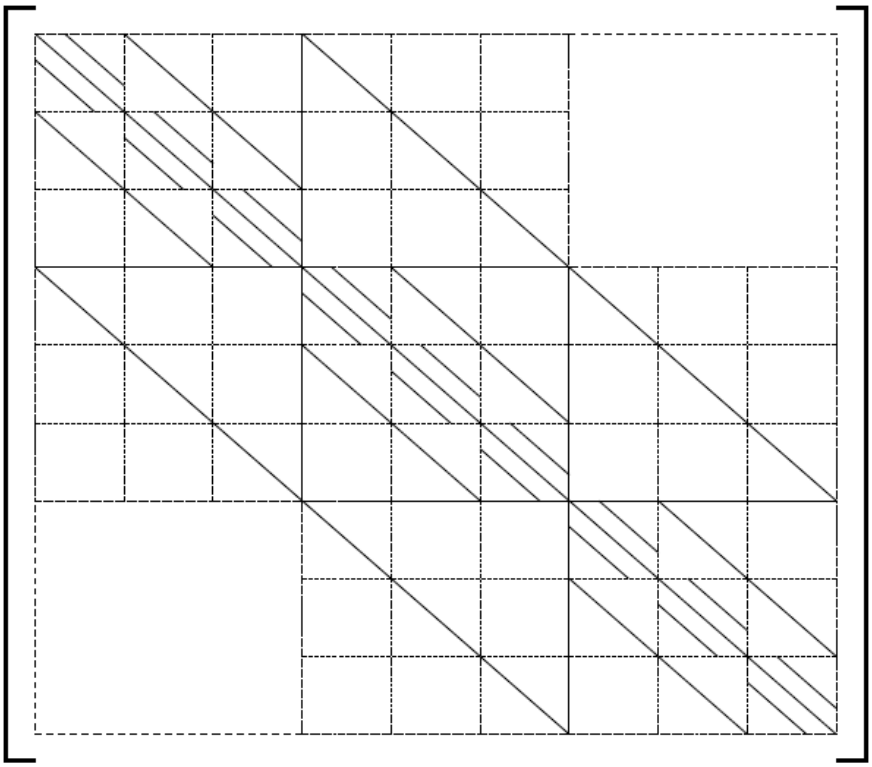

Figure 1: A Matrix representation

$$\left[d_1(p) = -\epsilon^x_{i+1/2,j,k}\frac{(h_{j-1}+h_j)\,(h_{k-1}+h_k)}{4h_i}\right] \tag{4}$$

- The second upper band diagonal which is shifted $N_x - 2$ columns from the first column

$$d_2(p) = \left[-\epsilon^y_{i,j+1/2,k}\frac{(h_{i-1}+h_i)\,(h_{k-1}+h_k)}{4h_j}\right] \tag{5}$$

- The third upper band diagonal which is shifted $(N_x - 2)(N_y - 2)$ columns from the first column

$$d_3(p) = \left[-\epsilon^k_{i,j,k+1/2}\frac{(h_{i-1}+h_i)\,(h_{j-1}+h_j)}{4h_k}\right] \tag{6}$$

The remaining elements of the upper triangular squared matrix A are set equal to zero. By symmetry we obtain the lower triagonal elements of the matrix A. Because the matrix A is sparse and large, we can implement efficient methods that optimally solve the linear system for U. Specifically, we use the biconjugate gradient stabilized method combined with the inexact LU decomposition of the matrix A. Having the numerical values for the nodes in the interior of the box, we finally add the previously removed prescribed values along the six faces to get the solution over the complete set of grid points.

Dirichlet Boundary Condition

The values of nodes $u_{1jk}, u_{N_x,j,k}, u_{i,1,k}, u_{i,N_y,k}, u_{ij1}$ and u_{ijN_z} along the six faces of the box is set to the values prescribed by a Debye-Hückel model for a multiple, non-interacting spheres with a point charges. The sphere radii are set to the atomic radii of the biomolecule and the sphere charges are set to the total charge of the protein.

Focus Boundary Condition

Our code uses linear interpolation to obtain the value of the potential at the six faces of the target box from the value of the potential obtained at larger domain.

Computational algorithm for Dirichlet boundary conditions

1. The code reads the (target) input file .inm to get the APBS input files (shifted dielectric coefficients, kappa function and pqr data file) as well as the number of (target) grid points, the box Lengths, temperature, and bulk properties (ionic strength and solvent dielectric coefficient) among other parameters.

2. The center of the grid is evaluated from the corresponding pqr file.

3. By using linear B-splines, the charge density is discretized to get f_{ijk} for $i = 1, .., N_x, \quad j = 1, .., N_y, \quad k = 1, .., N_z$. The Dirichlet boundary condition along the six faces of the box $u_{1jk}, u_{N_x,j,k}, u_{i,1,k}, u_{i,N_y,k}, u_{ij1}$ and u_{ijN_z} are calculated by using the temperature, the value of the bulk dielectric coefficient (usually water) and ionic strength.

4. The nonzero components of the matrix A, e.g., the diagonal elements $d_0(p), d_1(p), d_2(p)$, and $d_3(p)$, for $p = (k-2)(N_x-2)(N_y-2)+(j-2)(N_x-2)+i-1$ and $i = 2,..,N_x-1, \quad j = 2,..,N_y-1, \quad k = 2,..,N_z-1$ are evaluated by using the expressions (3), (4),(5), and (6). The values for the shifted dielectric coefficients and kappa function elements are obtained from the APBS input files. The values of the mesh size h_i, h_j and h_k are obtained from the number of grid points and the Length of the box. Next, the sparse upper triangular matrix A is constructed by filling with zeros the remaining elements of the matrix A. Next, the lower triangular elements of the matrix A are obtained by using the following symmetry property $A_{pq} = A_{qp}$ for $q = 1,..,(N_x-2)(N_y-2)(N_z-2)$ and $p = q,..,(N_x-2)(N_y-2)(N_z-2)$.

5. The elements of b_{ijk} are evaluated by using the values obtained for the discretized charge density f_{ijk} and the values of the Dirichlet boundary elements multiplied by the appropriate shifted dielectric coefficient values. The natural ordering $p = (k-2)(N_x-2)(N_y-2)+(j-2)(N_x-2)+i-1$ and $i = 2,..,N_x-1, \quad j = 2,..,N_y-1, \quad k = 2,..,N_z-1$ is used to construct the corresponding vector $b(p)$ (one index) from the data array structure (three indices) b_{ijk}.

6. The inexact LU decomposition of the matrix A is performed. The default tolerance value is set equal to 0.25 which provides a fast evaluation of the matrices L and U.

7. The resulting L and U matrices, the matrix A and the vector b are used to approximately solve $Au = b$ for the vector u using the biconjugate gradient stabilized method. The default accuracy is set equal to 10^-9 and the maximum number of iteration equal to 800.

8. The natural ordering relationship is used to convert the resulting vector $u(p)$ to data array structure to get the numerical solution for u_{ijk} for $i = 2,..,N_x-1, \quad j = 2,..,N_y-1, \quad k = 2,..,N_z-1$.

9. Finally the previously removed values of the nodes at the faces of the box are used to obtain the solution for the nodes u_{ijk} over the complete set of grid points, namely for $i = 1,..,N_x, \quad j = 1,..,N_y, \quad k = 1,..,N_z$.

10. The electrostatic potential u_{ijk} and the charge f_{ijk} maps are saved in dx format files.

11. The electrostatic potential surface $u_{ij(N_z+1)/2}$ is saved in tiff and fig format files for visualization purpose.

Computational algorithm for Focus boundary conditions

The algorithm reads the target input file finding that the boundary condition line says "focusname.inm" instead of "sdh". Then the matlab code automatically first reads that input file "focusname.inm" to solve the PB equation in the specified coarse grid using Dirichlet boundary condition as explained previously. It saves the resulting electrostatic potential solution in a temporary dx formatted file and then perform the following steps;

1. The code reads the (target) input file .inm to get the APBS input files (shifted dielectric coefficients, kappa function and pqr data file) as well as the number of (target) grid points, the box Lengths, temperature, and bulk properties (ionic strength and solvent dielectric coefficient) among other parameters.

2. The center of the grid is evaluated from the corresponding pqr file.

3. By using linear B-splines, the charge density is discretized to get f_{ijk} for $i = 1, .., N_x, \quad j = 1, .., N_y, \quad k = 1, .., N_z$. The Dirichlet boundary condition along the six faces of the target (smaller) box $u_{1jk}, u_{N_x,j,k}, u_{i,1,k}, u_{i,N_y,k}, u_{ij1}$ and u_{ijN_z} are calculated by using a three-linear interpolation for the electrostatic potential solution obtained previously at larger boxsides (Focus boundary Condition).

4. The nonzero components of the matrix A, e.g., the diagonal elements $d_0(p), d_1(p), d_2(p)$, and $d_3(p)$, for $p = (k-2)(N_x-2)(N_y-2) + (j-2)(N_x-2) + i - 1$ and $i = 2, .., N_x - 1, \quad j = 2, .., N_y - 1, \quad k = 2, .., N_z - 1$ are evaluated by using the expressions (3), (4),(5), and (6). The values for the shifted dielectric coefficients and kappa function elements are obtained from the APBS input files. The values of the mesh size h_i, h_j and h_k are obtained from the number of grid points and the Length of the box. Next, the sparse upper triangular matrix A is constructed by filling with zeros the remaining elements of the matrix A. Next, the lower triangular elements of the matrix A are obtained by using the following symmetry property $A_{pq} = A_{qp}$ for $q = 1, .., (N_x-2)(N_y-2)(N_z-2)$ and $p = q, .., (N_x-2)(N_y-2)(N_z-2)$.

5. The elements of b_{ijk} are evaluated by using the values obtained for the discretized charge density f_{ijk} and the values of the Dirichlet boundary elements multiplied by the appropriate shifted dielectric coefficient values. The natural ordering $p = (k-2)(N_x-2)(N_y-2)+(j-2)(N_x-2)+i-1$ and $i = 2,..,N_x-1, \quad j = 2,..,N_y-1, \quad k = 2,..,N_z-1$ is used to construct the corresponding vector $b(p)$ (one index) from the data array structure (three indices) b_{ijk}.

6. The inexact LU decomposition of the matrix A is performed. The default tolerance value is set equal to 0.25 which provides a fast evaluation of the matrices L and U.

7. The resulting L and U matrices, the matrix A and the vector b are used to approximately solve $Au = b$ for the vector u using the biconjugate gradient stabilized method. The default accuracy is set equal to 10^-9 and the maximum number of iteration equal to 800.

8. The natural ordering relationship is used to convert the resulting vector $u(p)$ to data array structure to get the numerical solution for u_{ijk} for $i = 2,..,N_x-1, \quad j = 2,..,N_y-1, \quad k = 2,..,N_z-1$.

9. Finally the previously removed values of the nodes at the faces of the box are used to obtain the solution for the nodes u_{ijk} over the complete set of grid points, namely for $i = 1,..,N_x, \quad j = 1,..,N_y, \quad k = 1,..,N_z$.

10. The electrostatic potential u_{ijk} and the charge f_{ijk} maps are saved in dx format files.

11. The electrostatic potential surface $u_{ij(N_z+1)/2}$ is saved in tiff and fig format files for visualization purpose.

Comment1: Note that in this case the user have to provide two inm.files and the corresponding dx and pqr files for both the coarse and target grids.

Comments2: In this version the user have to provide two pqr files, one representing the molecule by which the PB eq is solved and, the second one to define the center of the grid. It may be the same than the first one, but in general, for complex systems they are not.

Comments3: Fianlly, the user have to provide both directories for the input and output files respectively. In this way the user doesn't have to edit the source files at all. Just need

to provide the name of the input file and the full path as the only argument in the Matlab function MPABS (x).