Solving the Poisson-Boltzmann Equation with Mimetic Differences

Kim Ngan Huynh and Miguel A. Dumett

April 20, 2024

Publication Number: CSRCR2024-03

Computational Science & Engineering Faculty and Students Research Articles

Database Powered by the Computational Science Research Center Computing Group & Visualization Lab

COMPUTATIONAL SCIENCE & ENGINEERING

SAN DIEGO STATE UNIVERSITY

Computational Science Research Center
College of Sciences
5500 Campanile Drive
San Diego, CA 92182-1245
(619) 594-3430

© 2024

4 Results

4.1 Water Molecule

The known points for each atom of water generated from IQmol (Figure 2) and electrostatic potential net atomic charges generated from Q-Chem (Figure 3) are the initial conditions for the Poisson-Boltzmann algorithm. Most articles about solving nonlinear (1) have shown their roofs rather than their numerical solution plots. It is difficult to know what the solution plot of planar water molecule should look like. [2] only shows one solution figure but with a much larger superoxide dismutase (SOD) molecule. Figure 5 (left panel) is the solution plot of planar water molecule obtained initially. However, a significant amount of electrostatic potential is only observed at the position of Hydrogen atom, but not at the position of Oxygen atom. This becomes evident that this numerical solution is not correct.

The position along x-axis and y-axis of each atom, as provided by IQmol is observed to not coincide with the grid points. Upon adjusting these positions to align with the grid points, a distinct peak corresponding to the Oxygen atom becomes evident shown in Figure 5 (right panel). The observation of two positive single peaks at both Hydrogen atom locations and a negative single peak at the location of the Oxygen atom is consistent with the understanding of their respective charge distribution. The Hydrogen atoms have partial positive charges while the Oxygen atom has a negative partial charge.

Figure 5: Distribution of electrostatic potential around pre-shifted water molecule (*left*) and shifted water molecule (*right*)

4.2 Other Molecules

As testing the Poisson-Boltzmann algorithm with different initial conditions, three different molecules: HCl, SO_3, HCN are used. Due to the distance between two atoms in these molecules greater than 1, the solvent region is expanded to length of 8 and the radius of molecular region is increased to 2. All atoms of these molecules are shifted to be on the grid points. The correlation between the sign of the partial charge on an atom and the corresponding peak has shown in the solution plots below. A negative peak aligns with the position of an atom carrying a negative partial charge, and a positive peak aligns with the position of an atom carrying a positive partial charge.

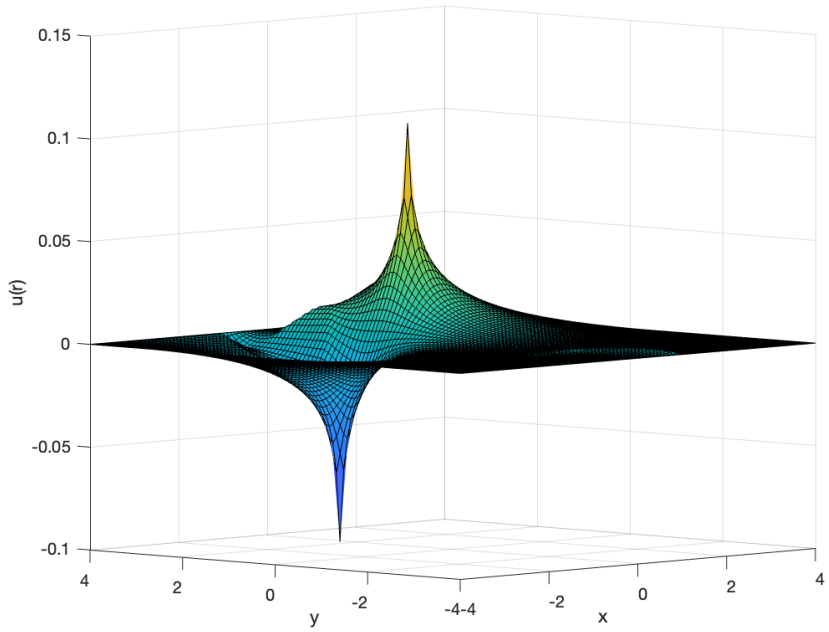

Figure 6: Distribution of electrostatic potential around HCl

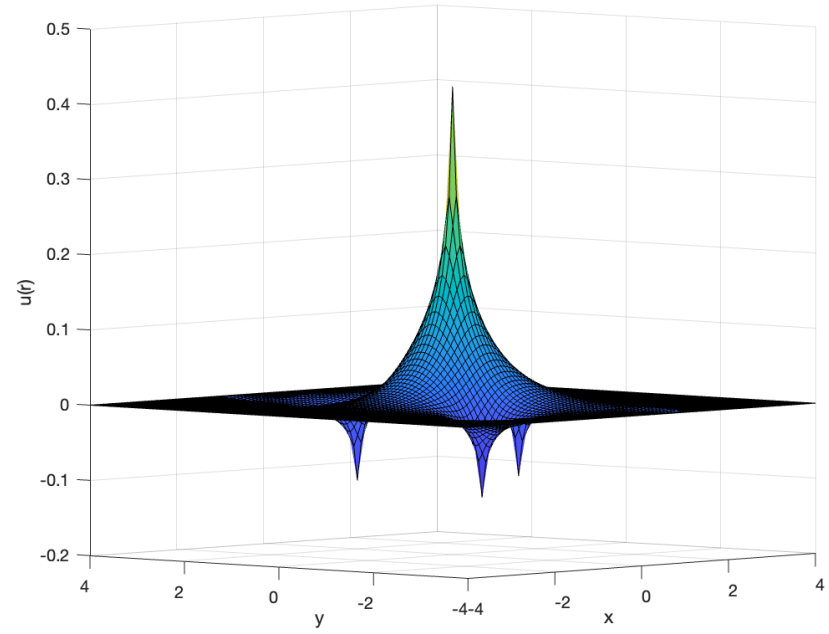

Figure 7: Distribution of electrostatic potential around SO_3

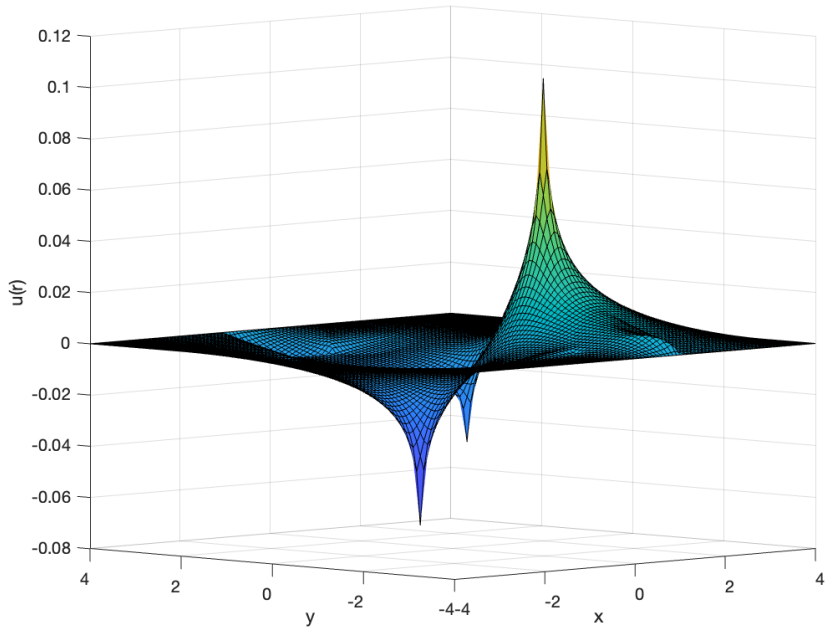

Figure 8: Distribution of electrostatic potential around HCN

Acknowledgements

I would like to thank Professor Yuezhi Mao for his knowledgeable advice on the chemistry aspect.

References

[1] J. Corbino and J. E. Castillo. "MOLE: Mimetic Operators Library Enhanced: The Open-Source Library for Solving Partial Differential Equations using Mimetic Methods". In: (2017). URL: https://github.com/csrc-sdsu/mole.

[2] M.J. Holst. *The Poisson-Boltzmann Equation Analysis and Multilevel Numerical Solution*. 1994. Chap. 1, pp. 1–14.

Appendix

This is input file of a planar water molecule (the positions of each atom are shifted to be on the grid points)

```
$molecule
0 1
  H    0.5000000    0.7400000    0.0000000
  O    0.0200000    0.0200000    0.0000000
  H    0.5000000   -0.7400000    0.0000000
$end

$rem
   BASIS  =  6-31g(D)
   ESP_CHARGES  =  TRUE
   GUI  =  2
   METHOD  =  B3LYP
   SCF_CONVERGENCE  =  8
$end
```

After submitting input file of water molecule to Q-Chem, Merz-Kollman electrostatic potential net atomic charges is obtained as the following

```
Merz-Kollman ESP Net Atomic Charges

    Atom                  Charge (a.u.)
----------------------------------------
    1 H                     0.426725
    2 O                    -0.846350
    3 H                     0.419625
----------------------------------------
Sum of atomic charges =     0.000000
```

The numerical solution of the Poisson-Boltzmann for a planar water molecule shown in Figure 5 (right) was obtained from the MATLAB code below

```
%Solving 2D Poisson-Boltmann Equation with mimetic operators
%PBE in Mimetic differences-D(epsilon(GU)) + (kappa)^2 sinh(U) = rho
addpath '/Users/lunahuynh/SDSU-23-24/COMP670/mole/mole_MATLAB'

%Parameters
k=2;            %order of accuracy
m=100;          %number of cells in x-axis
n=m;            %number of cells in y-axis
a=-2;           %west
b=2;            %east
c=-2;           %south
d=2;            %north
dx=(b-a)/m;
dy=(d-c)/n;

I=0.1;          %initial condition for ionic strength
C=7046.528885; %coefficient on the RHS of PBE

%Dielectric constant function epsilon
epsilon_s = 80;
epsilon_m = 2;

%Debye-Huckle constant function kappa
```

```
kappa_s = sqrt(I*8.486902807);
kappa_m = 0;

%2D staggered grid
xgrid = [a a+dx/2 : dx : b-dx/2 b];
ygrid = [c c+dy/2 : dy : d-dy/2 d];
[X, Y] = meshgrid(xgrid,ygrid);

%Initial location of each atom in water H20 (located on the grids)
%zi is Merz-Kollman ESP net atomic charges obtained from Q-Chem
x_H1 = 0.5000000;
y_H1 = 0.7400000;
zi_H1 = 0.426725;

x_O = 0.0200000;
y_O = 0.0200000;
zi_O = -0.846350;

x_H2 =  0.5000000;
y_H2 = -0.7400000;
zi_H2 = 0.419625;

x = [x_H1, x_O, x_H2];
y = [y_H1, y_O, y_H2];
z = [zi_H1, zi_O, zi_H2];

%RBF Interpolation (Gaussian RBF)
sigma = 0.0015;                  %parameter of the spread of the function
rbf = @(r) exp(-(r/(sigma)).^2); %r is a distance between 2 points

%RBF matrix (distance matrix)
R = zeros(length(x),length(x));
for i = 1:length(x)
    for j = 1:length(x)
    %for each pair of data points, calculate the Euclidean distance and apply rbf function
        R(i, j) = rbf(sqrt((x(i) - x(j))^2 + (y(i) - y(j))^2));
    end
end

%Solve linear system to find the weights for RBF interpolation
weights = R \ z';

%Interpolate at new points
Z = zeros(size(X));
for i = 1:numel(X)
    for j = 1:length(x)
        Z(i) = (Z(i) + weights(j) * rbf(sqrt((X(i) - x(j))^2 + (Y(i) - y(j))^2)));
    end
end

%RHS
rho = C.*Z;
rho = reshape(rho,[],1);

%Use PartitionOfUnity2DVectorFields for constant function epsilon
[epsilonh,epsilonv] = PartitionOfUnity2Dvector(4,1,epsilon_s,epsilon_m,m);
epsilonh = reshape(epsilonh,[],1);
epsilonv = reshape(epsilonv,[],1);
epsilon_hv = [epsilonh; epsilonv];
Epsilon = diag(epsilon_hv);

%Use PartitionOfUnity2DScalarFields for constant function kappa
kappa = (PartitionOfUnity2Dscalar(4,1,kappa_s,kappa_m,m).^2); %kappa^2
kappa = reshape(kappa,[],1);
Kappa = diag(kappa);

%Using the first term in the series expansion sinh(x) = x+x^3/3!+x^5/5!+...
%as an approximation of sinh(x). Therefore, mimetic differences would become
```

```
% –D(E (GU)) + K U = rho
% (–D(EG) + K) * U = rho
%Let A = (–D(EG) + K)
%             A * U = rho

%2D Mimetic gradient operator
G = grad2D(k,m,dx,n,dy);

%2D Mimetic divergence operator
D = div2D(k,m,dx,n,dy);

%Neumann BC
BC = robinBC2D(k, m, dx, n, dy, 0, 1);
A = –D*(Epsilon*G) + Kappa + BC;

% Solve for initial Ui
Ui = A \ rho;
Ui = reshape(Ui, m+2, n+2);
figure(1)
surf(X, Y, Ui);
xlabel('x');
ylabel('y');
zlabel('u_{i}(r)');

%Use Newton's Method to approximate a system of equations
%f(U) = –D(E(GU)) + K sinh(U) – rho
f = @(U) –D*(Epsilon*(G*U)) + Kappa*sinh(U) – rho;

Ui = reshape(Ui,[],1);
U = Ui;
maxIter = 100;
tolerance = 1e–15; %can be changed accordingly
iter = 0;

%Jacobian matrix is 1st partial derivatives of f with respect to U
J_linear = –D*(Epsilon*G);
J_nonlinear = diag(Kappa * cosh(U));
J = J_linear + J_nonlinear;
converge = false;
while iter < maxIter
    F = f(U);
    U_new = U – J\F;
    deltaU = U_new – U;
    if all(abs(deltaU) < tolerance) %check for convergence
        converge = true;
        break;
    end
    U = U_new;
    iter = iter + 1;
end

if converge
    disp(['Converged after ', num2str(iter), ' iterations.']);
else
    disp('Did not converge.');
end

U = reshape(U, m+2, n+2);
figure(2)
surf(X,Y,U);
xlabel('x');
ylabel('y');
zlabel('u(r)');
```

The input files of HCl molecule, SO_3 molecule, and HCN molecule are shown respectively below. Notice that all molecules are symmetrized and the atoms are shifted to be on the grid points

```
$molecule
0 1
Cl   -1.080000    1.4000000    0.0000000
H    0.3600000    1.3200000    0.0000000
$end

$rem
   BASIS  =  6-31g(D)
   ESP_CHARGES  =  TRUE
   GUI  =  2
   METHOD  =  B3LYP
   SCF_CONVERGENCE  =  8
$end
```

```
$molecule
0 1
S    0.0000000    0.0000000   -0.0000000
O    0.7678653   -1.3299818    0.0000000
O    0.7678653    1.3299818   -0.0000000
O   -1.5357307    0.0000000   -0.0000000
$end

$rem
BASIS  =  6-31g(D)
ESP_CHARGES  =  TRUE
GUI  =  2
METHOD  =  B3LYP
SCF_CONVERGENCE  =  8
$end
```

```
$molecule
0 1
C    0.0000000   -0.1005373    0.7324253
N    0.0000000    0.0206644   -0.7135753
H    0.0000000    0.9108826    1.1882759
$end

$rem
BASIS  =  6-31g(D)
ESP_CHARGES  =  TRUE
GUI  =  2
METHOD  =  B3LYP
SCF_CONVERGENCE  =  8
$end
```

The Merz-Kollman electrostatic potential net atomic charges of HCl,SO_3, and HCN molecule are listed in order below

Merz-Kollman ESP Net Atomic Charges

Atom	Charge (a.u.)
1 Cl	-0.247169
2 H	0.247169
Sum of atomic charges =	0.000000

Merz-Kollman ESP Net Atomic Charges

Atom	Charge (a.u.)
1 S	1.031620
2 O	-0.342988
3 O	-0.344316
4 O	-0.344316
Sum of atomic charges =	0.000000

Merz-Kollman ESP Net Atomic Charges

Atom	Charge (a.u.)
1 C	-0.109953
2 N	-0.146483
3 H	0.256436
Sum of atomic charges =	0.000000