Poisson's equation and Debye shielding

Debye shielding requires that we solve Poisson's equation and the Boltzmann relation. We will do this in one dimension. The values of the potential Φ will be assigned on grid points equally spaced in x. For example, Φ_n is the value of $\Phi(x)$ at the location x_n, the nth grid point. The first equation below is Poisson's equation written in the usual way. The second equation is the definition of the x derivative of Φ_n, using values from the grid. The third equation defines the second derivative of Φ_n. This is the d/dx operation done on $d\Phi_n/dx$, symmetrically about n. In the last equation, we have combined Eqs. 1 and 3 and have solved for Φ_n. We use the present values of Φ_{n+1} and Φ_{n-1} to calculate the second guess for Φ_n. The left side of the equation is the new guess based on the old guess which is on the right side. We will do this over and over again and keep getting better guesses. It is time to quit when the new guess is no longer much different from the old guess. n_0 is the zero order density, q is the charge, and T is the temperature in energy units.

$$-\frac{d^2}{dx^2}\Phi(x) = \left(n_i(x) - n_e(x)\right)q/\varepsilon_0$$

$$\frac{d\Phi}{dx} = \frac{1}{\Delta x}\left[\Phi_{n+1} - \Phi_n\right]$$

$$\frac{d^2\Phi}{dx^2} = \frac{1}{\Delta x^2}\left[\Phi_{n+1} - 2\Phi_n + \Phi_{n-1}\right]$$

$$\Phi_n = \frac{1}{2}\left[\Phi_{n+1} + \Phi_{n-1}\right] + \frac{1}{2}(\Delta x)^2\left[n_i(x_n) - n_e(x_n)\right]q/\varepsilon_0$$

Boltzmann relations:

$$n_e(x) = n_0 \exp[-q_e\Phi(x)/T]$$

$$n_i(x) = n_0 \exp[-q_i\Phi(x)/T]$$

Equations in dimensionless units

In the second line below we have multiplied or divided all the terms in Poisson's equation by constant values so that the last two bracketed terms are dimensionless. We then recognize that the first bracketed term is the Debye length squared and that this can be used to make the x derivatives dimensionless as well. The last three lines show the dimensionless versions of the variables.

$$-\frac{d^2}{dx^2}\Phi(x) = \left(n_i(x) - n_e(x)\right)q/\varepsilon_0$$

$$-\left(\frac{\varepsilon_0 T}{n_0 q^2}\right)\frac{d^2}{dx^2}\left(\frac{q\Phi(x)}{T}\right) = \left(\frac{n_i(x)}{n_0} - \frac{n_e(x)}{n_0}\right)$$

$$\frac{d^2}{d\tilde{x}^2}\tilde{\Phi}(\tilde{x}) = \tilde{n}_i(\tilde{x}) - \tilde{n}_e(\tilde{x})$$

$$\tilde{\Phi} = q\Phi/T$$

$$\tilde{x} = \frac{x}{\sqrt{\varepsilon_0 T/n_0 q^2}} = \frac{x}{\lambda_{Debye}}$$

$$\tilde{n} = n/n_0$$

In the pages below, we are using the dimensionless variables **without the tildes** on top, for convenience.

Δx is the spacing in the x direction. It is necessary to make this small enough to resolve details on the scale of the Debye length, so we will make $\Delta x = 0.5$. A vector (a matrix with one row) will contain the values of Φ. We will make the first and last values -1 and +1 and these will be the boundary conditions. The other values between will be set to zero at the beginning and then be improved by successive approximation. Our answer will be the potential profile between two plates biased to +1 and -1 in dimensionless units. The plates will be in a plasma with electron and ion Debye lengths of one.

$imax := 10$

$i := 0, 1 .. imax$

There will be 11 grid points.

$PhiGuess_i := 0$

Set them all to zero and then change the end values to +1 and -1.

$PhiGuess_0 := 1$ $PhiGuess_{imax} := -1$

$PhiGuess =$

	0
0	1
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	-1

We can change the number of points on the grid by increasing imax above.

The values for the potential will be saved in a matrix Phi defined in a PROGRAM LOOP.

The electron an ion densities will be given by $ni(\phi) := \exp(-\phi)$ $ne(\phi) := \exp(\phi)$

We will save all the values in a matrix. Each row in the matrix will be the next iteration.

$\Delta x := 0.5$ grid spacing in Debye lengths

$irows := 40$ number of iterations

$Phi :=$

$i \leftarrow 0$

for $j \in 0 .. imax$

$Q_{i,j} \leftarrow PhiGuess_j$

for $i \in 1 .. irows$

for $j \in 1 .. imax - 1$

$C \leftarrow \frac{Q_{i-1,j+1} + Q_{i-1,j-1}}{2} + \frac{1}{2} \cdot \Delta x^2 \cdot \left(ni\left(Q_{i-1,j}\right) - ne\left(Q_{i-1,j}\right) \right)$

$Q_{i,j} \leftarrow \frac{C + Q_{i-1,j}}{2}$

$Q_{i,0} \leftarrow Q_{i-1,0}$

$Q_{i,imax} \leftarrow Q_{i-1,imax}$

Q

Q stores the evolving solution.

This line averages the old values and new values for stability. See the comment on the fourth page.

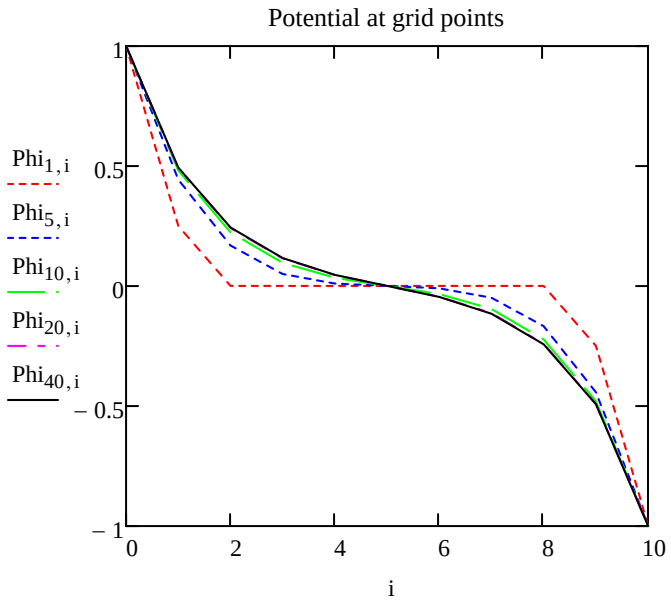
The final lines restore the boundary conditions.

This technique converges more rapidly if the new answer is averaged with the old answer before it is used. If the initial guess is too small, the second guess will "overshoot" and be too big, The averaging process helps convergence by averaging out the oscillations.

Phi =

	0	1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	0	0	0	0	0	0	-1
1	1	0.25	0	0	0	0	0	0	0	-0.25	-1
2	1	0.34	0.06	0	0	0	0	0	-0.06	-0.34	-1
3	1	0.39	0.11	0.02	0	0	0	-0.02	-0.11	-0.39	-1
4	1	0.42	0.14	0.03	$.91 \cdot 10^{-3}$	0	$.91 \cdot 10^{-3}$	-0.03	-0.14	-0.42	-1
5	1	0.44	0.17	0.05	$.76 \cdot 10^{-3}$	0	$.76 \cdot 10^{-3}$	-0.05	-0.17	-0.44	-1
6	1	0.46	0.19	0.06	0.02	0	-0.02	-0.06	-0.19	-0.46	-1
7	1	0.47	0.2	0.07	0.02	0	-0.02	-0.07	-0.2	-0.47	-1
8	1	0.47	0.21	0.08	0.03	0	-0.03	-0.08	-0.21	-0.47	-1
9	1	0.48	0.22	0.09	0.03	0	-0.03	-0.09	-0.22	-0.48	-1
10	1	0.48	0.22	0.1	0.03	0	-0.03	-0.1	-0.22	-0.48	-1
11	1	0.48	0.23	0.1	0.04	0	-0.04	-0.1	-0.23	-0.48	-1
12	1	0.49	0.23	0.1	0.04	0	-0.04	-0.1	-0.23	-0.49	...

If you scroll up and down the matrix you will see that the answers are all the same after about 30 iterations. At the top, you will see that the changes propagate toward the middle from the ends.



The 40th interation is the black line. The first iteration is the red line.

At the two boundaries, the potential decays exponentially.

Try it:
If above you change imax from 10 to 20 the curves become smoother.

Numerical instability uncovered: Suppose I didn't average the new answer with the old?

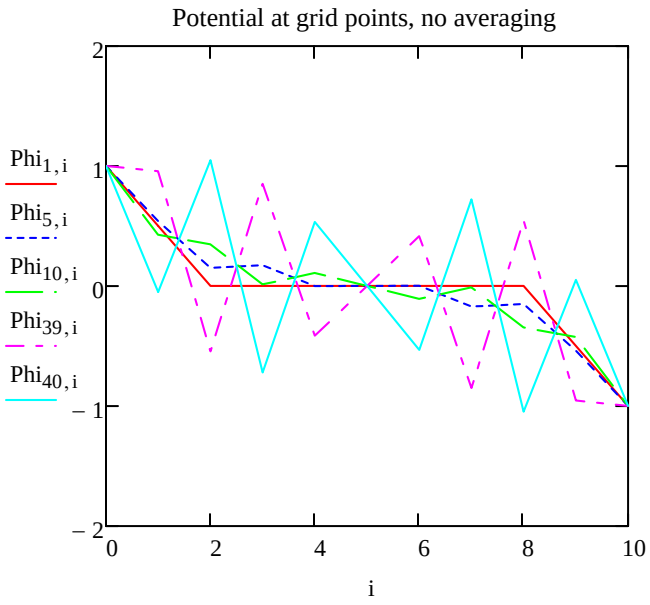
We will start over **without** averaging each interation with the previous one.

$PhiGuess2_i := 0$ $imax := 10$ $PhiGuess2_0 := 1$ $PhiGuess2_{imax} := -1$

$Phi :=$
- $i \leftarrow 0$
- for $j \in 0 .. imax$
  - $Q_{i,j} \leftarrow PhiGuess2_j$
- for $i \in 1 .. 40$
  - for $j \in 1 .. imax - 1$
    - $C \leftarrow \frac{Q_{i-1,j+1} + Q_{i-1,j-1}}{2} + \frac{1}{2} \cdot \Delta x^2 \cdot \left(ni\left(Q_{i-1,j}\right) - ne\left(Q_{i-1,j}\right) \right)$
    - $Q_{i,j} \leftarrow C$
    - $Q_{i,0} \leftarrow Q_{i-1,0}$
    - $Q_{i,10} \leftarrow Q_{i-1,imax}$
- Q

Note that the averaging is left out.

Note that iteration 39 and 40 are oscillating wildly about the right answer. Iteration 10 was actually better! If I had done 55 iterations, the program would have failed because the amplitude of the oscillations would have grown too large.


Potential at grid points, no averaging
Phi1, i
Phi5, i
Phi10, i
Phi39, i
Phi40, i
2
1
0
– 1
– 2
0
2
4
6
8
10
i