

# Application of finite difference method for reducing time complexity in potential calculation

Hun Seo

Department of Physics, Kyung Hee University

December 2016

## 1 Introduction

Most undergraduate general physics laboratory classes include experiments that measure potential distributions for specific charge distributions. Therefore, it is important to develop a method to obtain the theoretical value in order to confirm whether the experiment is performed properly. If you have a program that outputs the potential distribution quickly when you inputs the charge distribution, it will be useful as an educational point of view.

When the potential is given as a boundary condition, how to get the solution of the Poisson equation via the finite difference method has already been studied well. In this study, I will discuss the method of applying the finite difference method when we already knew about charge distribution instead of the boundary condition. In reverse I will get the appropriate boundary condition from the known charge distribution. I used version 2.7 of Python as a language, and I looked at ways to reduce error and computation time.

## 2 Composition of Program

### 2.1 Boundary Condition of Outskirt Border

In a vacuum at the two-dimensional non-continuous coordinate system, the Poisson equation is given as below;<sup>1</sup>

$$V(i, j) = \frac{1}{4}[V(i-1, j) + V(i+1, j) + V(i, j-1) + V(i, j+1) + \frac{\rho(i, j)h^2}{\epsilon_0}] \quad (1)$$

---

<sup>1</sup>Solving the Generalized Poisson Equation Using the Finite-Difference Method(FDM), James R. Nagel

in this equation  $i$  and  $j$  are value which shows how many pixels specific coordinate have moved each, while  $\rho$  and  $h^2$  are meaning charge density and pixel size.

It is recommended to have given boundary condition following outskirts of rectangle, for calculating approximate value acquired by finite-difference method(the potential given as a boundary condition will now be called as ‘boundary potential’), because when finite-difference method is being used inside, the potential of specific coordinate is determined by that of four neighboring coordinate’s. Also, it makes also possible to use a exact potential on one of four pixels during the following-the-border calculation, for both vertical and horizontal way(2 pixels for four each corners). Moreover, it will make all the coordinate’s potential converge evenly at the similar velocity.

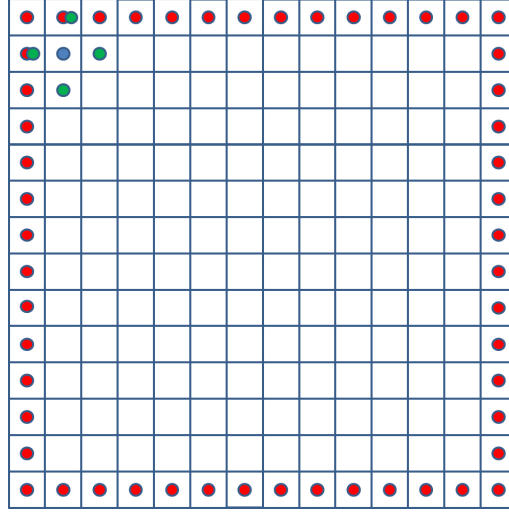


Figure 1: When using the FDM(finite-difference method) at near border, potential to be calculated will influenced evenly by boundary potential.

At Figure 1, red dots are potential which is given as boundary condition, while blue dots are position for observing the current potential. Green dots are potential which will be used to calculate the potential of blue dots. When potential are given on the outermost, as the Figure 1, potentials on each point are converge evenly to the exact potential value.

## 2.2 Boundary Condition of inside the border

At the point far from outskirts, potentials are converging relatively slow, since being influenced less by outskirts potential. For making boundary potentials to converge evenly inside the area, one can calculate the exact potential at the lattice points input as variable ‘distance’(hereafter ‘lattice point’), and then use as boundary potential (Figure 2). Actually, when using the lattice point, it

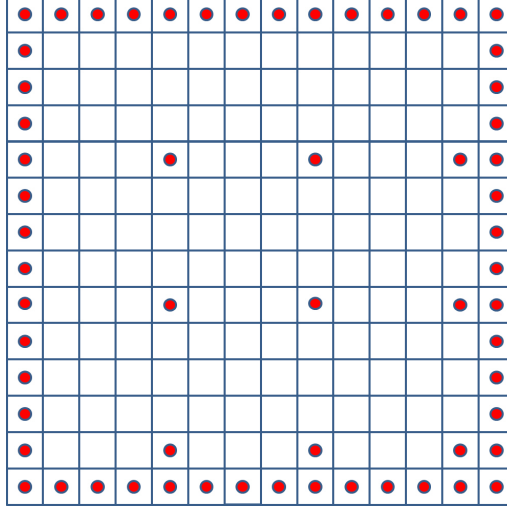


Figure 2: When distance is 4. Even at the point where far from boundary, it can be influenced enough by boundary potential.

converges faster to the exact potential comparing to the case not used. Potentials being used for boundary condition for lattice point will be gained precisely by summing  $1/r$  of all the electron, when  $r$  is regarded as a distance between position of potential observing and position of electron. Position of electron can be input by clicking the mouse or ‘drag and drop’.

### 2.3 The position of electron

What will happen if electron is at the exact point where one wants to observe the potential?  $1/r$  will diverge into infinity, since  $r$  becomes 0. For solving this, the electron will be regarded as on upper left vertex, not on the middle of pixels(Figure 3).

### 2.4 Simplification of unit

Though the potential being made by one electron is  $\frac{e}{4\pi\epsilon_0 r}$  in cgs(system of units), since amount I will actually calculate is  $\frac{1}{r}$ ,  $\frac{e}{4\pi\epsilon_0}$  will be regarded as  $1V \cdot h$  on  $h = 1unitlength$  for convenience. Charge density made by electron will give the value of charge density of  $\frac{e}{4h^2}$  for four each pixel in accordance with vertex electron positioned, which will make potential to be  $\frac{e}{4\epsilon_0}$ . Since regarded as  $\frac{e}{4\pi\epsilon_0} = 1V \cdot h$ , four each pixels will take the potential of  $\pi \cdot V$ . Thus, four pixels near the electron’s position would take a potential of  $\pi V$  for charge density  $\rho(i, j)$  on formula 1)

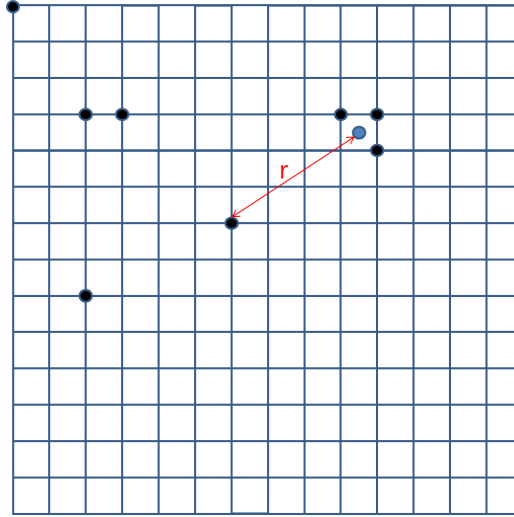


Figure 3: Even if the coordinates of the electrons indicated by the black dots overlap with the coordinates of which the potential is to be measured accurately, the positions are slightly shifted.

### 3 Execution of program

Once these assumptions are entered into the program, the program first obtains the potentials using a finite difference method (FDM). In most other studies using FDM, the difference between the potential obtained in the previous step and the potential obtained in the present step was approximated until the maximum value fell below a certain value. However, in this program, it is set up to calculate until the error ratio between the potential at the lattice point where the potential is accurately obtained and the potential obtained by the finite difference method becomes smaller than a certain ratio. As will be seen later, this method can be used to control the approximation of the error rate as a whole compared to the case of accurately finding the potential (DM). The variables we can control are the distance between the lattice points to be used as the boundary condition, the number and position of the electrons, the relaxation constant, and the horizontal to vertical length of the region.

#### 3.1 Relaxation Constant

In many cases, relaxation constants are used in finite difference methods. If the relaxation constant is introduced, the convergence speed into the precise potential is known to be faster (method of successive over-relaxation). If the relaxation constant  $w$  is introduced, equation 1 changes as follows.<sup>2</sup>

<sup>2</sup>James R. Nagel, "Solving the Generalized Poisson Equation Using the Finite-Difference Method(FDM)"(University of Utah:Department of Electrical and Computer Engineering,

$$V(i, j) = \frac{w}{4} [V(i-1, j) + V(i+1, j) + V(i, j-1) + V(i, j+1) + \frac{\rho(i, j)h^2}{\varepsilon_0}] - (1-w)V(i, j) \quad (2)$$

It is known that the optimal relaxation constant  $w$  is usually found by trial and error method, but especially when using FDM in the rectangular region, it is known as follows.

$$w = \frac{8 - \sqrt{64 - 16t^2}}{t^2} \quad (3)$$

In here  $t$  is

$$t = \cos \frac{\pi}{N_x} + \cos \frac{\pi}{N_y} \quad (4)$$

$N_x$  and  $N_y$  denote the width (number of pixels) in the horizontal direction and the height (number of pixels) in the vertical direction, respectively. The horizontal and vertical lengths of the windows used in the execution of the program are 450 pixels and 300 pixels, respectively, and the relaxation constant  $w = 1.982358$  in this case. We experimented whether this value is an optimal relaxation constant. The terms are described below, but the following results are obtained under the conditions of distance = 5, diserl = 15percentage, and number of electrons = 50.

Table 1: Relaxation Constant and Operation Time

Relaxation Constant	Overall average error rate	FDM Calculation time(sec)	DM Calculation time(sec)
1.982358	10.714	30	26
1.7	11.22	6	27
1.6	10.27	6	29
1.5	11.79	5	24
1.4	12.22	8	33

As shown in table 1, the optimal relaxation constant is around 1.5, and we will use relaxation constant  $w$  (relex in program) to 1.5.

### 3.2 Limit of Mean Error Rate at Lattice Points

As described earlier, FDM is set to approximate the error rate average at the lattice point until it becomes smaller than the input ratio (diserl in the code). But can we guarantee that the error at the lattice point is small and the error at the other points is small? In this section, we will look at how the average error

---

February 2012). p.7.

Table 2: Changes due to the variation of the lattice point error rate limit

lattice point error rate limit	Overall average error rate	FDM Calc time(sec)	DM Calc time(sec)
40	51.04	4	35
30	37.09	4	36
20	25.39	6	32
10	12.85	8	31
5	6.34	12	33
2	3.92	14	35

at the lattice point correlates with the mean error at all points, and how much computation time should be sacrificed to reduce the mean error at the lattice point. 50 electrons with arbitrary positions are in the region, and the distance is input as 5.

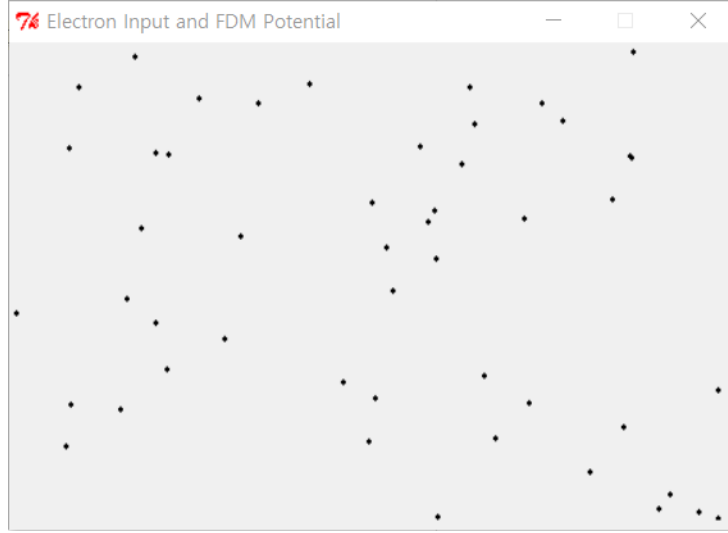


Figure 4: 50 charge distributions with random locations

As shown in Table 2, since the error of the lattice point is proportional to the average error rate of all the pixels, about 1.3 times, the desired error value can be reached by adjusting the desired lattice point error rate limit. Of course, in order to lower the error rate, more iteration is required, and the sacrifice of increasing computation time is inevitable. However, as we will see later, the overall error rate is also affected by the distance value, so you must first determine the appropriate distance value and then set the diserl value. The next section will look for the appropriate distance value.

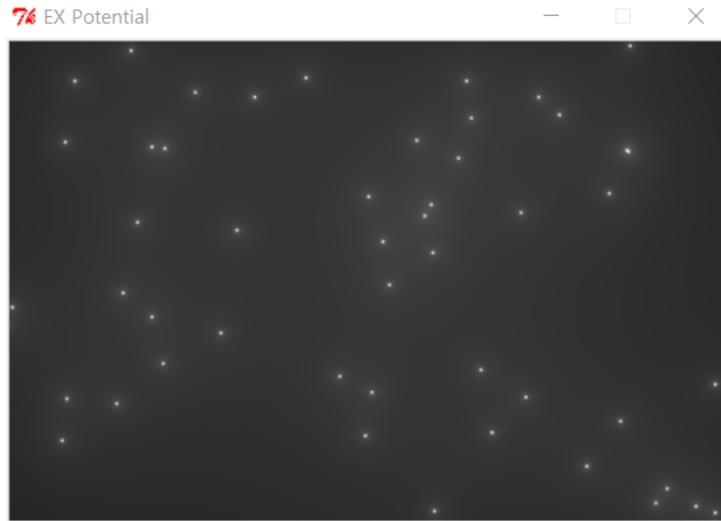


Figure 5: A potential distribution is obtained in the same charge distribution as Figure 4, and a white color is applied proportional to the potential.



Figure 6: When the lattice point error rate limit is 40 percent, the finite difference approximation result

### 3.3 Adjustment of lattice point spacing

The lattice point is a pixel that is used as a boundary potential in the region, and the distance between the lattice points is input as a *distance*. When using

lattice points, the computation speed, that is, the convergence speed is faster than when not used, and therefore a smaller error rate can be reached in a short time. As the lattice point spacing becomes smaller and dense, the number of lattice points increases and therefore more boundary conditions can be used, but instead the time to calculate the correct potential at the lattice point is slowed down. We then experimented with the optimal distance value. Likewise, the number of electrons was 50, and  $\text{diserl} = 10\text{percentage}$ .

Table 3: The error rate and the computation time according to the variation of the lattice point interval

lattice point interval	Overall average error rate	FDM Calc time(sec)	DM DM Calc time(sec)
25	20.90	196	31
20	20.58	97	32
15	18.76	55	29
10	16.64	26	29
5	12.87	8	32
3	8.67	13	35
2	4.99	21	34

When distance = 25, it required 196 iterations to satisfy the error rate limit. The grid spacing suitable for the conditions of this study is thought to be about 5. When the lattice point interval is 5, the number of lattice points to calculate the potential directly within the region is about  $1/25$  of the total pixels.

### 3.4 Number of electrons and computation time

If you look at the number of electrons, you need to calculate  $1/r$  as the number of electrons to calculate the potential at one observation point in the method of finding the potential directly. Therefore, the amount of computation increases in proportion to the number of electrons. However, in the finite difference approximation method, since only the potentials at the outermost boundary and the inner lattice point need to be accurately calculated, the computation time is not greatly influenced by the number of electrons. Table 4 shows the variation of the computation time according to the number of electrons generated at arbitrary positions under the condition that the distance between the lattice points = 5 pixels and the mean error limit at the lattice point ( $\text{diserl}$ ) = 10percentage.

As shown in Table 4, the computation time is not significantly affected by the number of electrons in FDM. However, in the method of directly obtaining the electrons, the computation time is proportional to the number of electrons as expected. The point that we can see in this section is that it is advantageous to use the finite difference method rather than the direct method of obtaining the electric potential as the number of electrons increases.



Table 4: Number of electrons and computation time

of electrons	FDM Calc time(sec)	DM Calc time(sec)	Overall error rate limit
50	10	19	12.92
100	7	30	12.88
150	8	48	12.87
200	9	59	12.82
250	10	71	12.78

## 4 Conclusion

If the number of electrons is small, a direct method of adding  $1/r$  is a good method that does not have any error and computation time is not long. However, as we have seen in this study, if the number of electrons is large (about 300 or more), finding the potential using the finite difference method greatly reduces the computation time. In this study, the boundary potentials at the lattice points (spaced apart from each other) within the outer region and the inner region of the rectangular shape region were determined. Assuming it was good, I tried to find the boundary potential. In Section 3.4, we confirmed that this method can reduce the computation time significantly.

## References

- [James R. Nagel(2012)] James R. Nagel, "Solving the Generalized Poisson Equation Using the Finite-Difference Method(FDM)"(University of Utah:Department of Electrical and Computer Engineering, February 2012).