Lab 6: Electrostatics

SI1136 Simulation and Modeling

Axel Strömberg

December 6, 2018

Contents

6.1

6.1 Project i

Numerical solution of the potential within a rectangular region.[1]

a) Determine the potential $V(x, y)$ in a square region with linear dimension $L = 10$. The boundary of the square is at a potential $V = 10$. Choose the grid size $\Delta x = \Delta y = 1$. Before you run the program, guess the exact form of $V(x, y)$ and set the initial values of the interior potential $10\,\%$ lower than the exact answer. How many iterations are necessary to achieve $1\,\%$ accuracy? Decrease the grid size by a factor of two, and determine the number of iterations that are now necessary to achieve $1\,\%$ accuracy.

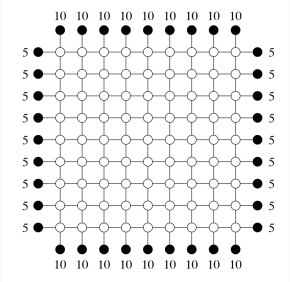
6.1 Project ii

b) Consider the same geometry as in part **a)**, but set the initial potential at the interior sites equal to zero except for the center site whose potential is set equal to four. Does the potential distribution evolve to the same values as in part **a)**? What is the effect of a poor initial guess? Are the final results independent of your initial guess?

c) Modify the program so that the value of the potential at the four sides are

$$V = 5, 10, 5, 10$$

respectively as shown in the figure. Sketch the equipotential surfaces. What happens if the potential is 10 on three sides and 0 on the fourth? Start with a reasonable guess for the initial values of the potential at the interior sites and iterate until 1 % accuracy is obtained.



[1] Problem 10.10 in H. Gould - An Introduction to Computer Simulation Methods

The exact solution

The exact solution of the boundary problem

$$\begin{cases} \nabla^2 V(x,y) = 0 \\ V(0,y) = V(11,y) = V(x,0) = V(x,11) = 10 \\ 0 \leq x, y \leq 11 \end{cases}$$

is the trivial solution

$$V(x,y) = 10 \ .$$

a) Firstly, the initial guess is set to 90 % of the exact solution. For example, when $n = 10$:

$$V = \begin{array}{c} i \backslash j \\ 0 \\ 1 \\ 2 \\ \vdots \\ 8 \\ 9 \\ 10 \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & \cdots & 8 & 9 & 10 \end{array} \\ \begin{bmatrix} 0 & 10 & 10 & \cdots & 10 & 10 & 0 \\ 10 & 9 & 9 & \cdots & 9 & 9 & 10 \\ 10 & 9 & 9 & \cdots & 9 & 9 & 10 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 10 & 9 & 9 & \cdots & 9 & 9 & 10 \\ 10 & 9 & 9 & \cdots & 9 & 9 & 10 \\ 0 & 10 & 10 & \cdots & 10 & 10 & 0 \end{bmatrix} \end{array}$$

6.1 Method ii

Secondly, the solution is iterated until the tolerance is met for the grid sizes $n = 10$ and $n = 5$. This implies $9^2 = 91$ and $4^2 = 16$ points to iterate over. The tolerance criterion is defined as

$$\max_{1 \leq i,j \leq n} \left| \frac{V(i,j) - V_{\text{exact}}(i,j)}{V_{\text{exact}}(i,j)} \right| < 0.01 \ .$$

The field is then iterated for $n = 2, 3, 4, \ldots, 51$. The number of steps required is plotted to the grid size together with a log-log plot to illustrate the quadratic dependence.

b) The same method as in **a)** was used. Here however, the initial guess was set to

$$V = \begin{bmatrix} 0 & 10 & \cdots & 10 & 0 \\ 10 & 0 & \cdots & 0 & 10 \\ \vdots & \vdots & 4 & \vdots & \vdots \\ 10 & 0 & \cdots & 0 & 10 \\ 0 & 10 & \cdots & 10 & 0 \end{bmatrix}$$

for even n

and

$$V = \begin{bmatrix} 0 & 10 & \cdots & 10 & 0 \\ 10 & 0 & \cdots & 0 & 10 \\ \vdots & \vdots & \begin{matrix} 4 & 4 \\ 4 & 4 \end{matrix} & \vdots & \vdots \\ 10 & 0 & \cdots & 0 & 10 \\ 0 & 10 & \cdots & 10 & 0 \end{bmatrix}$$

for odd n.

c) The same method as in **a)** was used. Here however, the grid size was set to $n = 100$ and the initial guesses was set to

$$V = \begin{bmatrix} 0 & 10 & \cdots & 10 & 0 \\ 5 & 7.5 & \cdots & 7.5 & 5 \\ \vdots & \vdots & 7.5 & \vdots & \vdots \\ 5 & 7.5 & \cdots & 7.5 & 5 \\ 0 & 10 & \cdots & 10 & 0 \end{bmatrix}$$

for the first boundary condition

and

$$V = \begin{bmatrix} 0 & 10 & \cdots & 10 & 0 \\ 10 & 9.9 & \cdots & 0.1 & 0 \\ \vdots & \vdots & 5.0 & \vdots & \vdots \\ 10 & 9.9 & \cdots & 0.1 & 0 \\ 0 & 10 & \cdots & 10 & 0 \end{bmatrix}$$

for the second boundary condition.

Secondly, instead of solving the problems analytically and comparing the result to the exact solution, the tolerance criterion was defined as

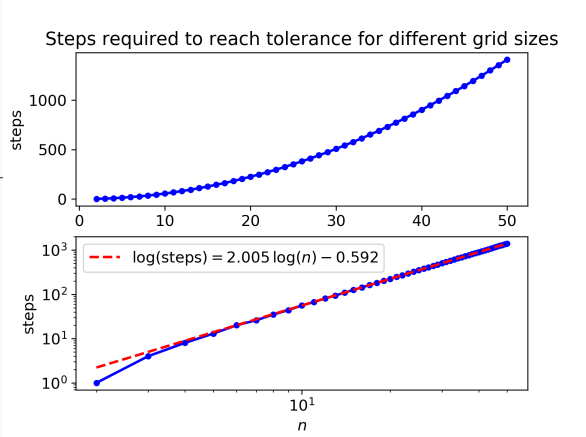
$$\max_{1 \leq i,j \leq 100} \left| \frac{V_n(i,j) - V_{n-1}(i,j)}{V_{n-1}(i,j)} \right| < 0.0001 \ .$$

That is, when two following iteration had a maximum difference of 0.01 % at any point the iteration was stopped. Three plots are generated:

1. The potential is plotted as a color gradient.
2. The equipotential surfaces are added with $\Delta V = 0.5$ step increments.
3. Potential labels are added to the equipotential lines.

a) For large grid sizes, the steps required is proportional to the area of the grid.

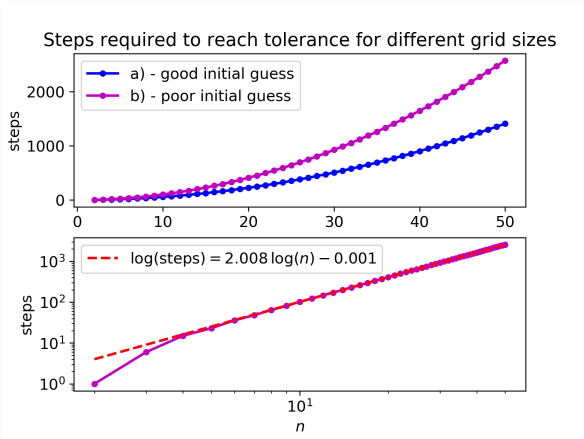
grid size, n	steps required
5	13
10	56
n	$0.55n^2, \quad n \gg 1$



6.1 Results ii

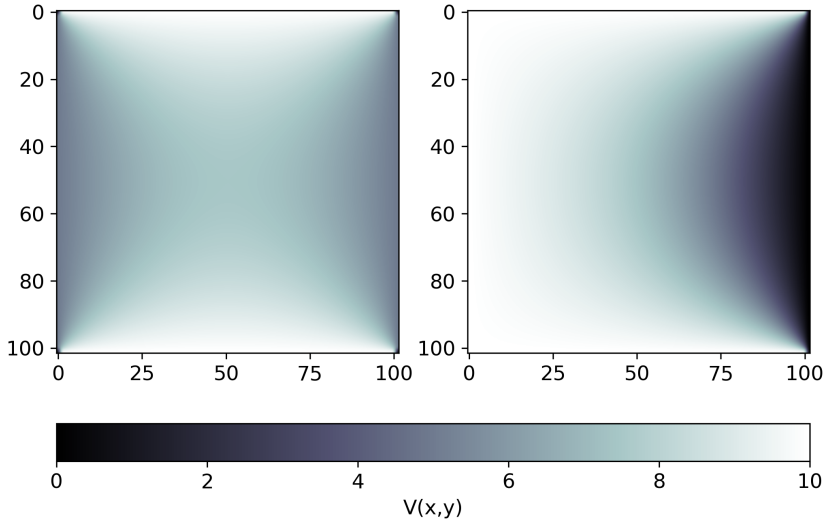
b) For large grid sizes, the steps required is proportional to the area of the grid. The number of steps was increased in comparison to the better initial guess.

grid size, n	steps required
5	23
10	102
n	$n^2, \quad n \gg 1$

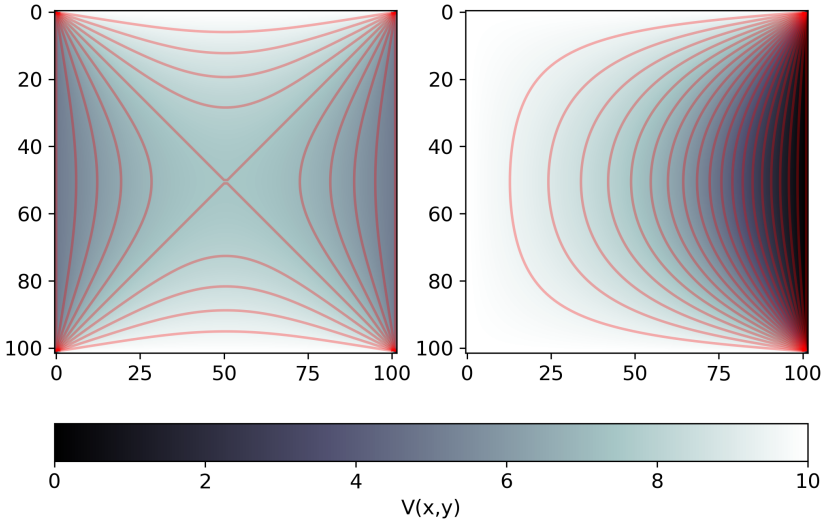


6.1 Results iii

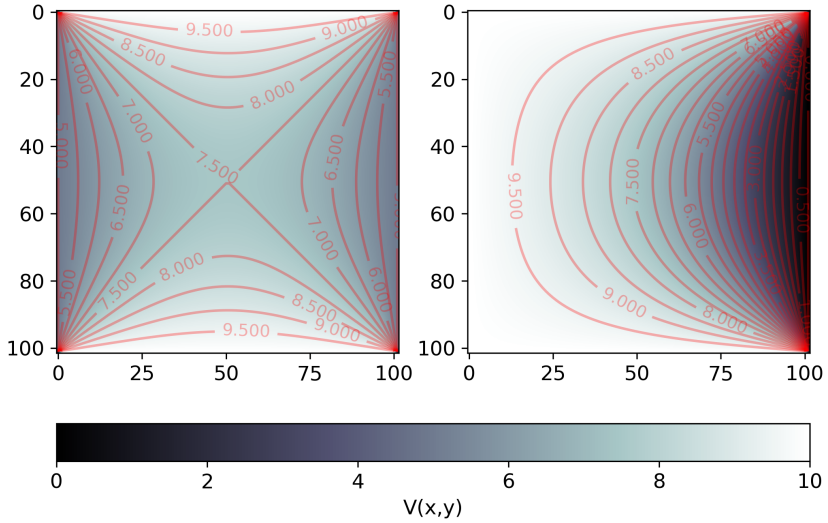
c) 1.



c) 2.



c) 3.


0
20
40
60
80
100
0
25
50
75
100
9.500
9.000
8.500
8.000
7.500
7.000
6.500
6.000
5.500
0
2
4
6
8
10
V(x,y)

6.2

Gauss-Seidel relaxation.[2]

a) Modify the program used in project **6.1** so that the potential at each site is updated sequentially. That is, after the average potential of the nearest neighbor sites of site i is computed, update the potential at i immediately. In this way the new potential of the next site is computed using the most recently computed values of its nearest neighbor potentials. Are your results better, worse, or about the same as for the simple relaxation method?

b) Imagine coloring the alternate sites of a grid red and black, so that the grid resembles a checkerboard. Modify the program so that all the red sites are updated first, and then all the black sites are updated. This ordering is repeated for each iteration. Do your results converge any more quickly than in part **a)**?

[2] Problem 10.11 in H. Gould - An Introduction to Computer Simulation Methods

6.2 Method i

a) Initially, the relax method was updated. Now the calculated values was stored directly to the V-matrix after each calculation of V_{ij} instead of updating the V-matrix after V_{ij} is calculated for all $i, j \in [1, n]$.
Then, the initial guess is set to 90 % of the exact solution. For example, when $n = 10$.

$$V = \begin{array}{c} i \backslash j \\ 0 \\ 1 \\ 2 \\ \vdots \\ 8 \\ 9 \\ 10 \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & \cdots & 8 & 9 & 10 \end{array} \\ \begin{bmatrix} 0 & 10 & 10 & \cdots & 10 & 10 & 0 \\ 10 & 9 & 9 & \cdots & 9 & 9 & 10 \\ 10 & 9 & 9 & \cdots & 9 & 9 & 10 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 10 & 9 & 9 & \cdots & 9 & 9 & 10 \\ 10 & 9 & 9 & \cdots & 9 & 9 & 10 \\ 0 & 10 & 10 & \cdots & 10 & 10 & 0 \end{bmatrix} \end{array}$$

Lastly, the solution is iterated until the tolerance is met for the grid sizes in the range $n = 2, 3, 4, \ldots, 51$. The tolerance criterion is defined as

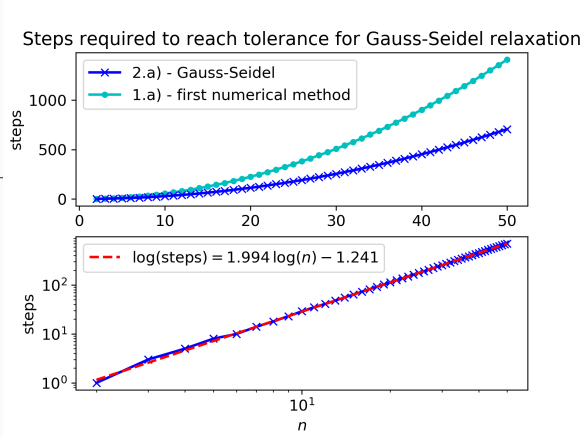
$$\max_{1 \leq i,j \leq n} \left| \frac{V(i,j) - V_{\text{exact}}(i,j)}{V_{\text{exact}}(i,j)} \right| < 0.01 \ .$$

The number of steps required is plotted to the grid size together with a log-log plot to illustrate the quadratic dependence.

b) The same method as in **a)** was used. Here however, the calculation was made with the checker-relaxation method.

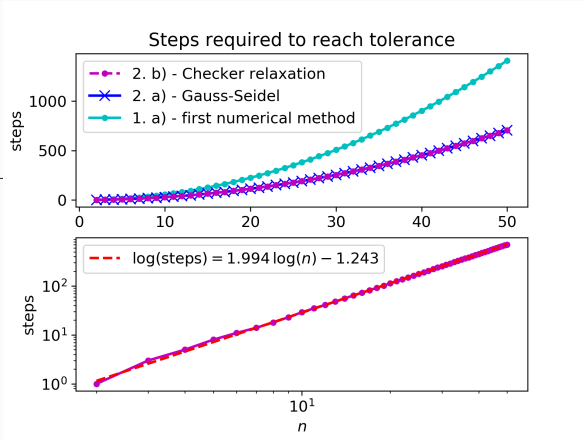
a) The method was improved in comparison to the methods previously used.

grid size, n	steps required
5	8
10	29
n	$0.29\ n^2, \quad n \gg 1$



b) The number of steps exactly the same as for the Gauss-Seidel relaxation, only different on a few grid sizes.

grid size, n	steps required
5	8
10	29
n	$0.29n^2, \quad n \gg 1$



6.3

6.3 Project i

The multigrid method.[3]

a) Write a program that implements the multigrid method using Gauss–Seidel relaxation on a checkerboard lattice as in project **6.2 b)**. In its simplest form the program should allow the user to intervene and decide whether to go to a finer or coarser grid, or to remain at the same level for the next relaxation step. Have the program print the potential at each site of the current level after each relaxation step. Test your program on a 4×4 grid whose boundary sites are all equal to unity, and whose initial internal sites are set to zero. Make sure that the boundary sites of the coarser grids are also set to unity.

6.3 Project ii

b) The exact solution for part **a)** gives a potential of unity at each point. How many relaxation steps does it take to reach unity within 0.1 % at every site by simply using the 4×4 grid? How many steps does it take if you use one coarse grid and continue until the coarse grid values are within 0.1 % of unity? Is it necessary to carry out any fine grid relaxation steps to reach the desired accuracy on the fine grid? Next start with the coarsest scale, which is just one site. How many relaxation steps does it take now?

c) Repeat part **b)** but change the boundary so that one side of the boundary is held at a potential of 0.5. Experiment with different sequences of prolongation, restriction, and relaxation.

[3] Problem 10.26 in H. Gould - An Introduction to Computer Simulation Methods

6.3 Method i

a) The multigrid method is implemented and tested for a 4×4-grid. The boundary conditions was set to $V(x, y) = 1$ on the border.

b) The method in **6.2 b)** was repeated for a 4×4-grid. The boundary condition was set to $V{=}1$ on the border. The initial guess was set to 0 everywhere. This was then compared with the method in **6.3 a)**.

6.3 Method ii

c)
1. The grid was initially set to the size of $n = 2$ and the center value was set as the average of the of the border values, 0.875.
2. The grid size was doubled to $n = 4$.
3. The new, fine grid locations that have two closest neighbors from the coarse grid was calculated as the average of the two.
4. The average of the rest was calculated as the average of the 4 closest neighbors.
5. The checker relaxation was then repeated until a maximum of $1\,\%$ differed on any point between two iterations.
6. Step 2-5 was repeated until desired grid size.

a) The method gave exact results for all $n \in \left\{ n = k^2 \middle| k \in \mathbb{N} \right\}$.

The first iteration gave the V-matrix

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

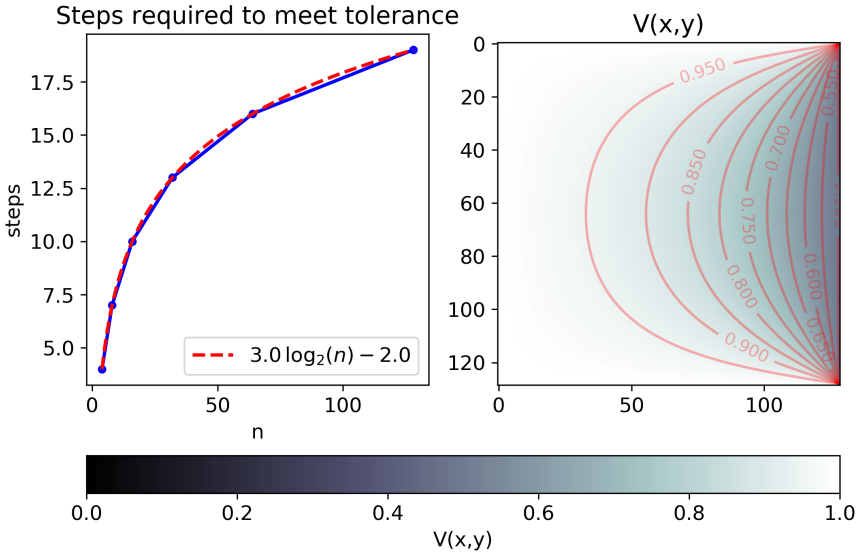
The second iteration gave the V-matrix

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

6.3 Results ii

b)
- It takes 11 relaxation steps to reach tolerance for a 4×4-grid using the checker-relaxation.
- By using the method in **6.3 a)** it takes 2 relaxation steps.
- Since the boundary condition is constant at the border it is not necessary to carry out any more than one fine grid relaxation steps to reach the desired accuracy on the fine grid.
- The coarsest scale, $n = 2$, was already implemented in **6.3 a)**. The method required 2 relaxation steps.

c)


Steps required to meet tolerance
steps
n
$3.0\log_2(n) - 2.0$
V(x,y)
0.950
0.850
0.750
0.800
0.700
0.600
0.900
V(x,y)

Appendix

Source code

Available at:

https://github.com/axelstr/SI1336-Simulations-and-Modeling/tree/master/6%20Electrostatics