

PDEs: Boundary Problems and Relaxation

Aleksandar Ivanov

May 2, 2021

1 Problem Statements

Problem 1

Find the Poiseuille coefficient C for the flow of a viscous fluid through a pipe with the cross-section pictured in fig. 1. In nondimensional form this is given by

$$\nabla^2 u = -1 \quad (1)$$

$$C = \frac{8\pi}{S^2} \iint u \, dx \, dy, \quad (2)$$

where S is the area of the shape and u is the velocity. Test the benefits of the Successive Over-Relaxation (SOR) and Chebyshev-accelerated SOR methods and find the optimal parameters for both. Be careful with edges that don't exactly fall on the discretized points of the mesh.

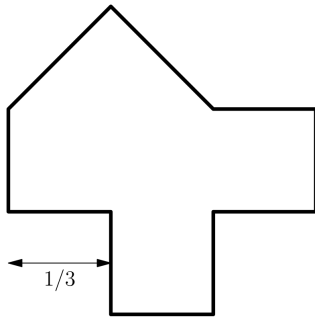


Figure 1: Cross-sectional profile of the pipe that we want to simulate the flow in.

Problem 2

We heat a cylinder with dimensions $h = 2r$ with a constant flux on its lower base. We cut the side of the cylinder into two pieces along a plane passing through its center and perpendicular to the bases

and keep these two pieces at different constant temperatures. The upper base is also at a constant temperature and this temperature is the same as the temperature of the lower half of the side. Find the temperature profile for the described situation. What would be different if we isolated the upper base instead?

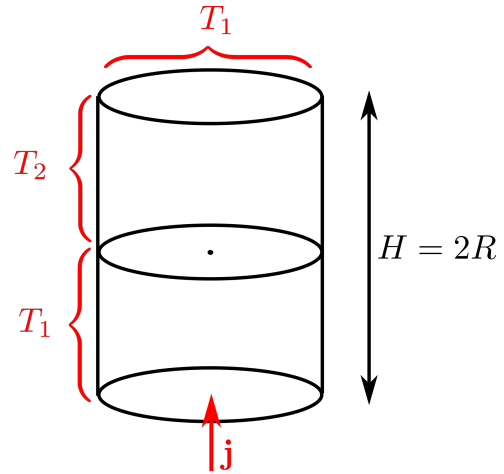


Figure 2: Geometry and boundary conditions of the second problem.

2 Numerical Methods

To make life simpler our domain will always be the unit square with an appropriate mask so that we're never solving outside of the shape we choose. Furthermore, we'll use a uniformly spaced mesh of points in both the x and y directions with the same step size h .

Throughout, we will be making use of four iterative methods. The simpler two are the Jacobi and Gauss-Seidel methods. These are just discretization of the second derivative with the only differ-

ence between them being where we update from. The Jacobi method updates its values completely from the previous iteration while the Gauss-Jacobi method updates its points from both the previous and the current iteration if they have already been calculated.

With these two the Poisson equation $\nabla^2 u = q$ becomes

$$u_{j,k}^{n+1} = \frac{1}{4} \left(u_{j-1,k}^n + u_{j,k+1}^n + u_{j+1,k}^n + u_{j,k-1}^n \right) - \frac{1}{4} q_{j,k} h^2 \quad (3)$$

for the Jacobi method and

$$u_{j,k}^{n+1} = \frac{1}{4} \left(u_{j-1,k}^{n+1} + u_{j,k+1}^n + u_{j+1,k}^n + u_{j,k-1}^{n+1} \right) - \frac{1}{4} q_{j,k} h^2 \quad (4)$$

for the Gauss-Seidel method with the usual discretization notation employed.

The two other methods we will be using are improvements on top of the Gauss-Seidel method. These are the Successive Over-Relaxation (SOR) method and the Chebyshev symmetrized version of SOR. Both of them introduce the parameter ω , which is used to take a weighted average of the newly calculated solution and the previous solution. Namely,

$$u_{j,k}^{n+1} = u_{j,k}^n + \omega(u_{j,k}^{n+1*} - u_{j,k}^n), \quad (5)$$

where $u_{j,k}^{n+1*}$ is the solution calculated by Gauss-Seidel. This new parameter can in theory take values in the range $0 < \omega < 2$ and still converge, however, in practice for our type of problem, values higher than 1 are the ones to look at. The characteristic property of SOR is that for a specific value of ω the algorithm converges much faster than the unmodified Jacobi and Gauss-Seidel algorithms. This optimal ω_0 depends on the number of points N and also has a weak dependence on the shape we choose. For a square with a uniform mesh as described above we have a theoretical approximation to the optimal value given by

$$\omega_0 = \frac{2}{1 + \sin(\pi/N)}. \quad (6)$$

Finding the more exact value will be something we're interested in.

The Chebyshev symmetrized version [2] uses the same parameter, but it has a different order of updating the values. It updates the values in a chess-board pattern so that whenever we update a cell we're using its neighbors from the previous iteration and not a mix of the previous and current iterations.

It's also possible to implement the Chebyshev algorithm such that it doesn't require a value of ω upfront, but instead it also updates the value of ω as we go through the iterations starting from the neutral $\omega = 1$.

To determine when to stop the calculation we will use the condition that the 2-norm of the difference of the last two solutions per cell is smaller than a given tolerance

$$\|u^{n+1} - u^n\|_2 / N^2 < \epsilon. \quad (7)$$

As the boundary of our region of interest can be arbitrary, it's likely that it won't fall exactly on the established mesh of points. This problem can be avoided in two ways. Either we make the density of points large enough for the effect of the boundary to not matter. Or we can modify the differential equation by making the last step inside the domain smaller than the rest so that we end up exactly on the boundary [3]. The last point on the inside has four neighbors, at $(j-1, k)$, $(j, k+1)$, $(j+1, k)$ and $(j, k-1)$, and in general we have to look at modifying all of them. If we denote the last step size by $\alpha_b h$, $b \in \{0, 1, 2, 3\}$, $\alpha_b < 1$, one alpha for each of the four directions in the order above, then we get the modified Laplacian formula

$$\nabla^2 u \approx \sum_b \frac{2u_b}{\alpha_b(\alpha_b + \alpha_{b'})} - 2 \left(\frac{1}{\alpha_0 \alpha_2} + \frac{1}{\alpha_1 \alpha_3} \right) u_{j,k}, \quad (8)$$

where $b' = \text{mod}(b+2, 4)$, u_b denotes the value of the function at the interpolated boundary point closest to the neighbor denoted by b . What the above formula is telling us is how to modify the coefficients in the difference formula to take into account the smaller steps and that this change affects opposite points in pairs, an expected fact just from the definition of the derivative.

In the case of Dirichlet boundary conditions the above u_b are 0 and the formula is simpler. This however doesn't mean that there is no effect because the coefficient in front of $u_{j,k}$ is now different.

For the second problem we also have boundary conditions of Neumann-like type. To apply these we will discretize the derivative in the asymmetric form

$$\left. \frac{du}{dx} \right|_{j,k} \approx \frac{u_{j+1,k} - u_{j,k}}{h}. \quad (9)$$

Until now, we haven't talked about the initial guess for the solution needed to start iterating. This is because in the case of Dirichlet conditions we can always start with the initial guess being 0 everywhere and iterate from there.

In the case of Neumann-like conditions this is not the case since the condition fixes the difference of neighboring points at the boundary but not their actual value. This means that if we start with a wrong value at that boundary, it may happen that we can never correct it.

However, in the case of part 1 of Problem 2 we can avoid this problem because there the top and right boundaries in the $r-z$ plane are still given by Dirichlet conditions, so that if we start integrating from those boundaries we can expect the solution to be correct. In the case of part 2 of Problem 2 we can't avoid it so easily, and we have to be more careful in choosing the initial guess.

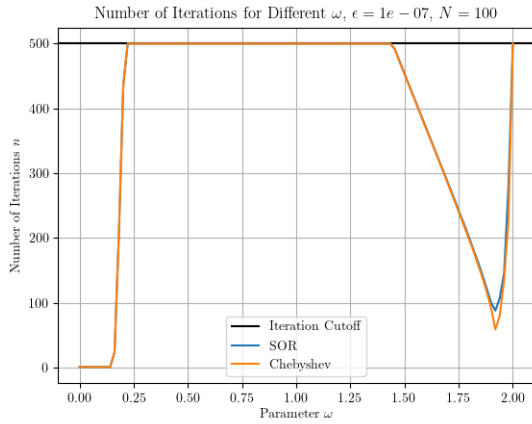


Figure 3: Number of iterations necessary for convergence as a function of ω . The iterations have been limited to 500 to save on unnecessary computation. The shape of the region is the square.

3 Results: Problem 1

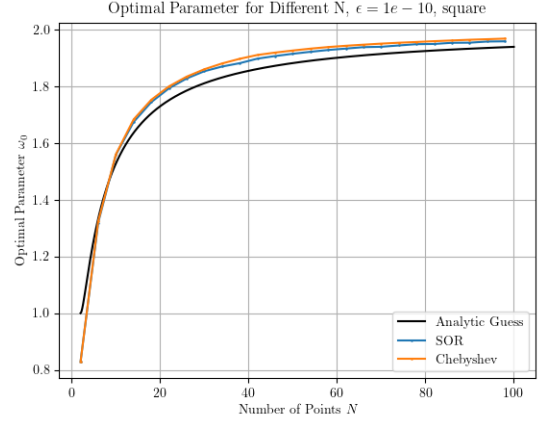


Figure 4: Optimal parameter ω as a function of the number of points N . The shape of the region is the square.

To speed up all subsequent calculations we will first look at finding the optimal ω_0 . First we look for it by fixing the number of points to $N = 100$. By looking around the theoretical prediction we find that near the optimal value SOR and Chebyshev converge after less than 500 iterations. Knowing this we can do a sweep across the whole region of ω and plot the number of iterations needed for convergence for each one. Furthermore, we limit the maximal number of iterations to 500 because we know that the value we're looking for has less than that. Since SOR's convergence is very dependent on ω , this saves a lot of unnecessary computation. This procedure is shown in fig. 3. As expected from the theoretical approximation, the optimal value is near $\omega = 1.93$. The optimal values from this figure are

$$\begin{aligned} \text{SOR} : \omega_0 &= 1.948 \pm 0.001 \\ \text{Chebyshev} : \omega_0 &= 1.957 \pm 0.001. \end{aligned} \quad (10)$$

The values for SOR and Chebyshev are similar but not exactly the same and furthermore, Chebyshev does slightly better at its optimal value than SOR. We also see a fall to 0 iterations at values of ω near 0. This doesn't represent any real solution and is an artifact of the stopping condition. At small ω we're updating the solution so little that the condition

for convergence gets fulfilled even through we don't have an actual solution.

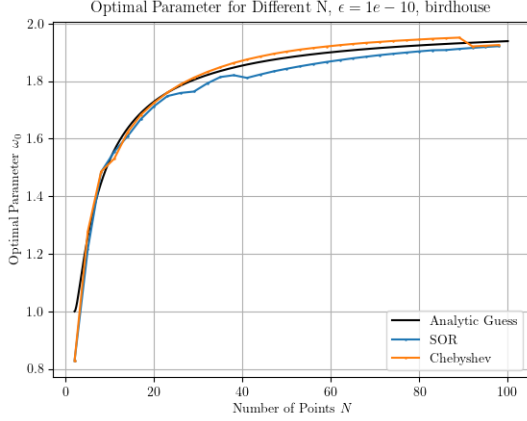


Figure 5: Optimal parameter ω as a function of the number of points N . The shape of the region is the birdhouse.

Next we do a more thorough search for different N . For the unmodified square this dependence is shown in fig. 4. We see that the analytical solution is quite a good guess for the optimal values, although there are deviations. We also see as before that the Chebyshev optimal value is slightly higher than the SOR optimal value. The optimal value is

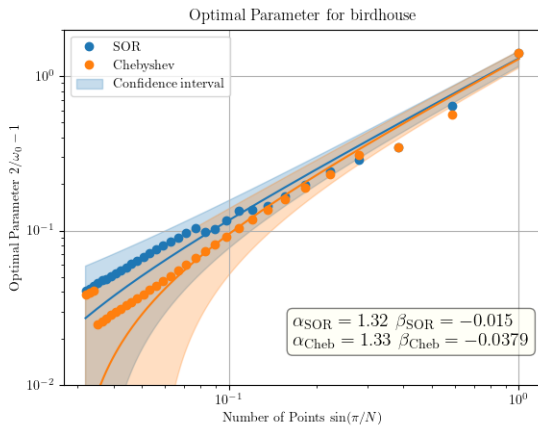


Figure 6: Fitting the $\alpha - \beta$ modified model for the birdhouse-shaped region using both the SOR and Chebyshev methods.

also dependent on the shape of our domain. If we try to capture the behavior of the optimal value as a modification to eq. (6) we can try to fit the data with the following model

$$\frac{2}{\omega_0} - 1 = \alpha \sin\left(\frac{\pi}{N}\right) + \beta, \quad (11)$$

where we have introduced the fitting parameters α and β . Figure 5 shows the dependence for our birdhouse-shaped domain. Here we see that the analytical guess is again good however the SOR and Chebyshev lines are now farther apart.

A better perspective is obtained through fig. 6, which shows a fit of the $\alpha - \beta$ modified model above with the results from the Chebyshev and SOR methods. Since the model has inverse values of our quantities which have a more uniform distribution, we show the plot in a log-log scale. We see that the linear model describes the data well enough and gives the fitting parameters as

$$\text{SOR} : \alpha = 1.32 \pm 0.06, \quad \beta = -0.02 \pm 0.01$$

$$\text{Cheb.} : \alpha = 1.33 \pm 0.07, \quad \beta = -0.04 \pm 0.02. \quad (12)$$

where the error only accounts for the statistics of the data and not further model error.

Having found the optimal values for running SOR and Chebyshev, we continue with testing their performance. First we're interested in how fast

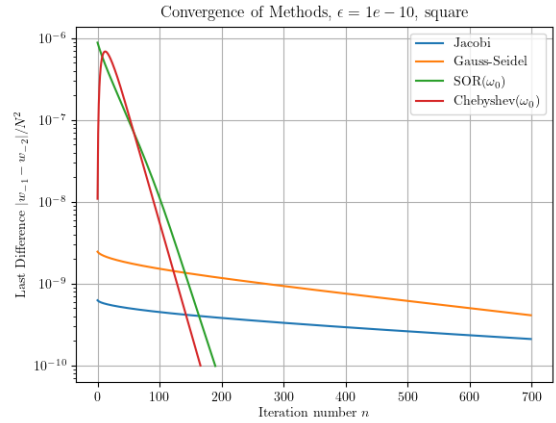


Figure 7: Speed of convergence of the four methods on the square domain. The simple Jacobi and Gauss-Seidel methods don't even converge at this tolerance and maximal number of iterations.

they converge. Figure 7 shows the speed of convergence of the four methods we’re considering. It’s a plot of the deviation measure used in the stopping condition as a function of the number of iterations. It has been capped at $n = 700$ iterations and in that time the Jacobi and Gauss-Seidel methods did not even converge to the given tolerance $\epsilon = 10^{-10}$. The SOR and Chebyshev methods, on the other hand, with the optimal values of ω converged very fast. Chebyshev converged the fastest of the four. More concerning is the fact that the Jacobi and Gauss-Seidel methods’ slope on this plot is very small, meaning that even if we were to wait longer they wouldn’t be that closer to converging.

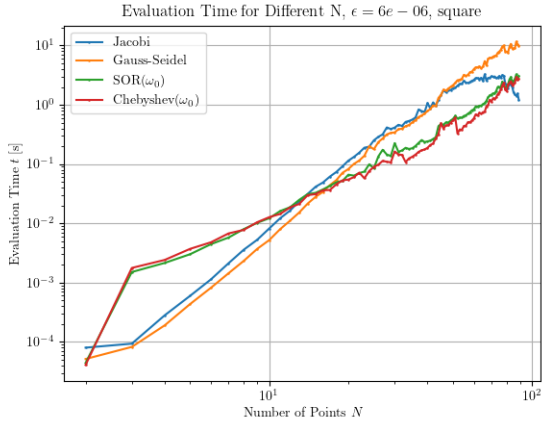


Figure 8: Evaluation time of the four methods on the square domain. Again, the SOR and Chebyshev methods do better.

In a similar vein, we also have how long it takes for the methods to find the solution, i.e. the evaluation time. This is shown in fig. 8. Of course, the SOR and Chebyshev methods again do better asymptotically (in this case meaning from ~ 10 points). At large values there seems to be an unexpected speed-up of the Jacobi method, but upon further inspection this turns out to also be an artifact of the stopping condition and the method does not actually converge to the solution.

A fit of these data gives the asymptotic power law dependence of the evaluation time on the number

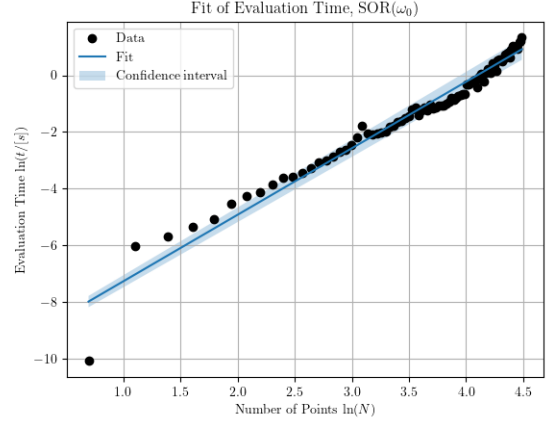


Figure 9: Evaluation time fit for the SOR method on the square domain. We see evidence that the asymptotic behavior is going to have a larger power.

of points as

$$\begin{aligned} \text{Jacobi, Gauss - Seidel : } & t \sim N^{\approx 3.3} \\ \text{SOR, Chebyshev : } & t \sim N^{\approx 2.3}, \end{aligned} \quad (13)$$

however here we have tested up to $N = 100$ and further testing with higher values is probably necessary to determine these numbers more accurately. Figure 9 shows the fit for the SOR method as corroboration that the evaluation time indeed has a power law dependence. At the same time we see evidence that for higher N the power will be higher, so we can’t claim to have reached the asymptotic regime.

Having tested some of the properties of these methods in regard to convergence and speed we continue with actually solving for the flow profile and the Poiseuille coefficient of the birdhouse-like shape. Some flow profiles including the birdhouse-shaped one can be seen in figs. 10 to 15 all of which have been calculated with the Chebyshev method. The circle serves as a good test since its coefficient is exactly 1 in theory. The astroid does a good job of showing off why we need to take better care of the boundary because even after implementing the above boundary interpolation we can still see that one of its contours (shown dashed) is still not particularly smooth.

After implementing the algorithms, generating flow profiles is as easy as creating the proper mask.

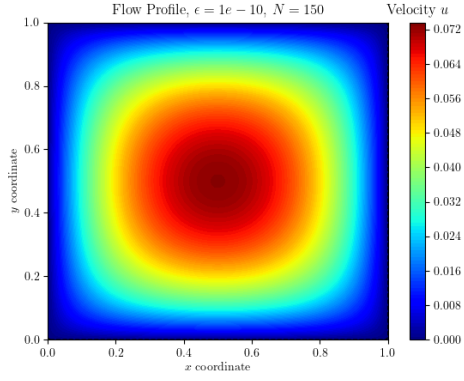


Figure 10: Flow for square profile.

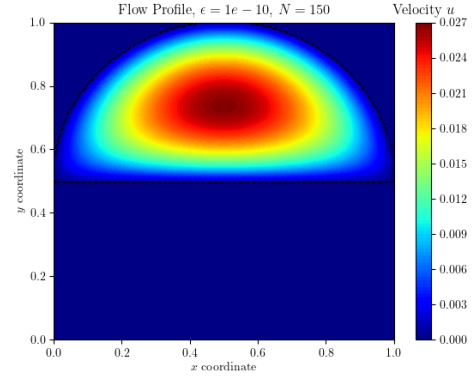


Figure 13: Flow for semicircular profile.

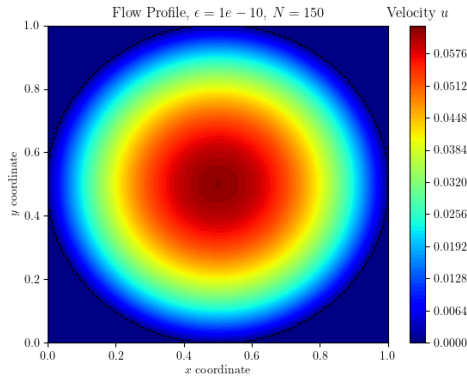


Figure 11: Flow for circle profile.

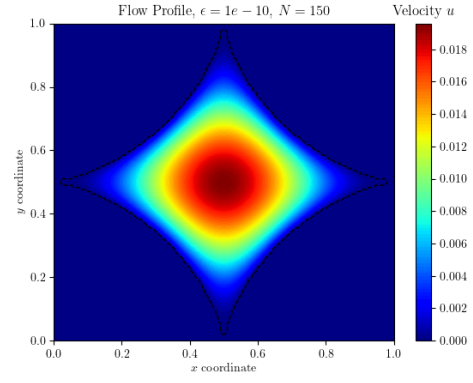


Figure 14: Flow for astroid profile.

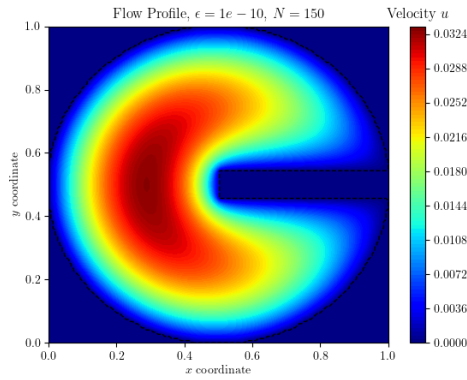


Figure 12: Flow for circle with bar profile.

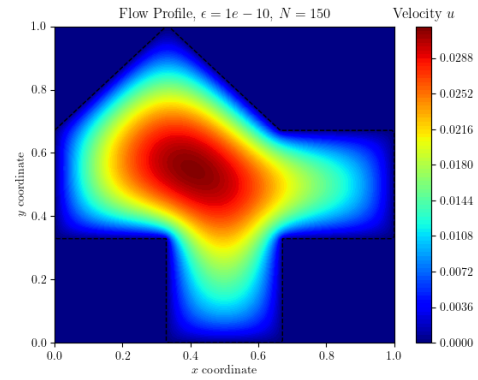


Figure 15: Flow for birdhouse profile.

| Shape | C_{calc} | C_{real} |
|------------|-------------------|-------------------|
| square | 0.866 | |
| circle | 0.992 | 1.000 |
| circle-bar | 0.456 | |
| semicircle | 0.773 | 0.757 |
| astroid | 0.755 | |
| birdhouse | 0.593 | |

Table 1: Poiseuille coefficient values for the different shapes calculated with $N = 200$. The real values are provided when possible.

The Poiseuille coefficients for these profiles are given in table 1. Two of the coefficients — for the circle and semicircle — are also compared to their real values from where we can see that our values are not magnificently correct. For the circle we get it right to two decimal place, while for the semicircle we only get one decimal place. Thus, we shouldn't be too trusting of these values; all we can say with certainty is that for the birdhouse-like shape it's ~ 0.6 . Very spikey shapes like the astroid are even more suspicious.

4 Results: Problem 2

Having already written the algorithms and discussed the differences in this second problem. All that is needed is to adapt to the Neumann-like conditions when necessary.

The three free parameters of the problem statement — T_1 , T_2 and j — are actually two independent, non-trivial free parameters because we can always choose a different 0 for the temperature. We'll choose to set $T_1 = 0$.

The boundary conditions specified in the problem statement have to be supplemented with another boundary condition at $r = 0$. The one that physically makes sense here is

$$\left. \frac{\partial T}{\partial r} \right|_{r=0} = 0, \quad (14)$$

which can be seen by imagining a Gaussian cylinder around the z -axis and letting its radius go to 0.

Since all the boundary conditions are now fixed, we can construct the solutions. Some profiles and their parameter values are shown in what follows. (figs. 16 to 21)

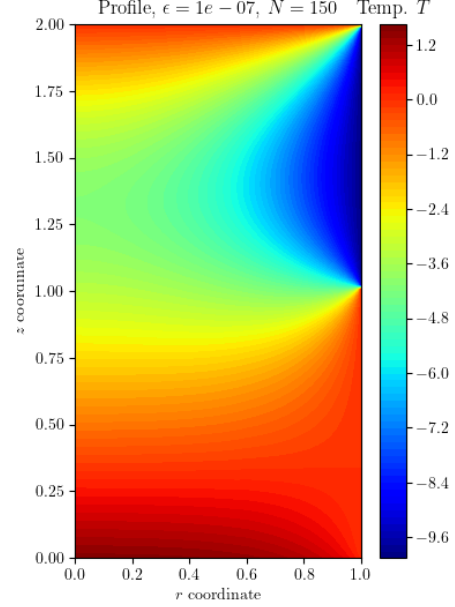


Figure 16: Temperature profile for $T_2 = -10$, $j = -10$.

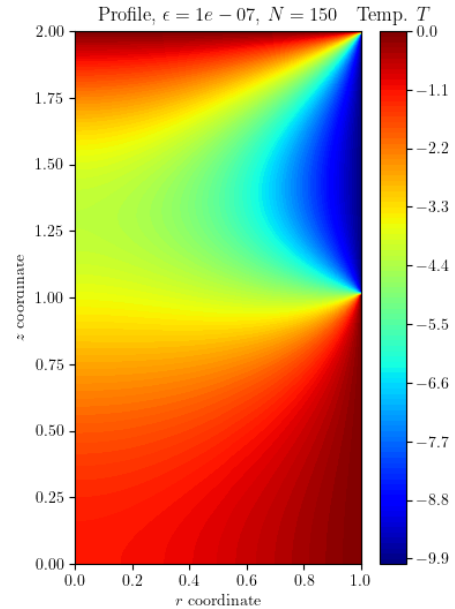


Figure 17: Temperature profile for $T_2 = -10$, $j = 0$.

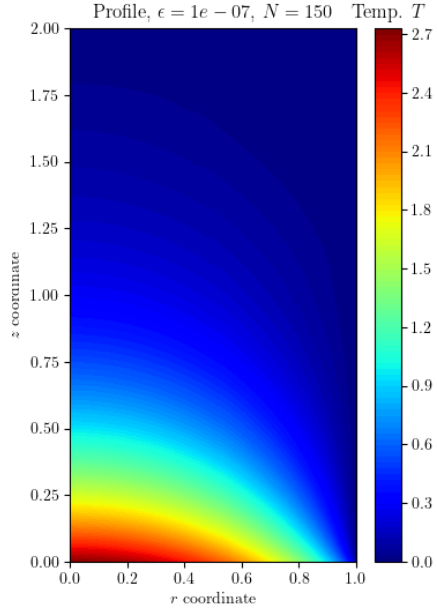


Figure 18: Temperature profile for $T_2 = 0$, $j = -10$.

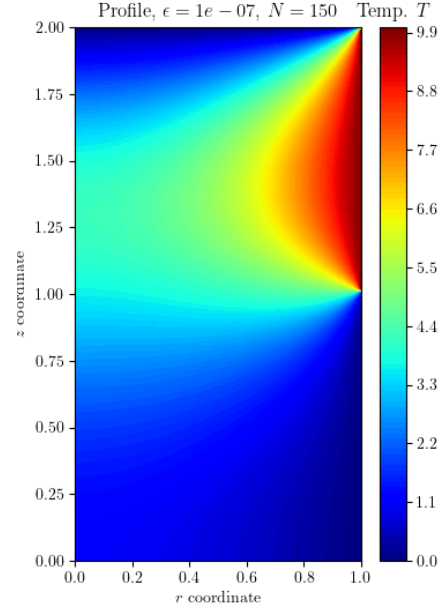


Figure 20: Temperature profile for $T_2 = 10$, $j = 0$.

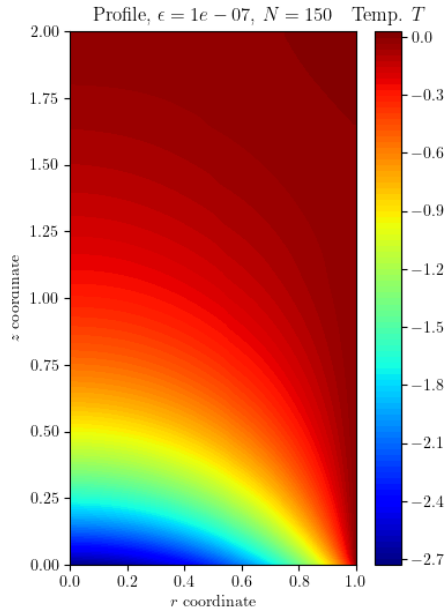


Figure 19: Temperature profile for $T_2 = 0$, $j = 10$.

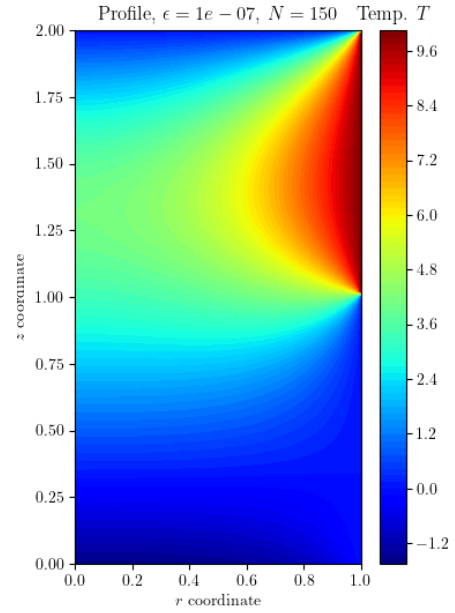


Figure 21: Temperature profile for $T_2 = 10$, $j = 10$.

Some expected behavior that we can see in the figures is that we always have the contour lines perpendicular to the z -axis, which is a consequence of the boundary condition at $r = 0$. Similarly, when $j = 0$ the contours are perpendicular to the r -axis. When the flow j and temperature T_2 have the same sign and are not 0 a relatively flat contour can appear where the effects of the two balance (figs. 16 and 21). If $T_2 \neq 0 (= T_1)$ then the profile has two humps; one dominated by the temperature T_1 , the other by the temperature T_2 .

For part two, as discussed before, we need to be more careful about the initial conditions and more vigilant of the final solution to be sure that we have the correct one.

A boundary condition for an isolated side is simply a special case of the one for a constant current

$$\left. \frac{\partial T}{\partial z} \right|_{z=H} = 0, \quad (15)$$

so that now we have Neumann-like conditions on three of the sides.

Again a smattering of temperature profiles with different parameter values using this new boundary condition is shown in...

A lot of the same observations from before can also be seen here. However, some new things can be seen as well. Comparing fig. 24 and fig. 25, we can see that, as an effect of the constant temperature reservoirs, changes are somewhat localized to their halves of the cylinder; the profile at the top doesn't change that much even when the bottom changes considerably.

With this setup, the top doesn't share a reservoir with a part of the side and consequently the pictures are more symmetric than in part 1.

References

- [1] J. Kiefer. "Sequential minimax search for a maximum". In: *Proc. Amer. Math. Soc.* 4 (1953), pp. 502–506. DOI: <https://doi.org/10.1090/S0002-9939-1953-0055639-3>.
- [2] Zheng Li, Changjun Li, and David J. Evans. "Chebyshev acceleration for SOR-like method". In: *International Journal of Computer Mathematics* 82.5 (2005), pp. 583–593. DOI: [10.1080/00207160512331331129](https://doi.org/10.1080/00207160512331331129).

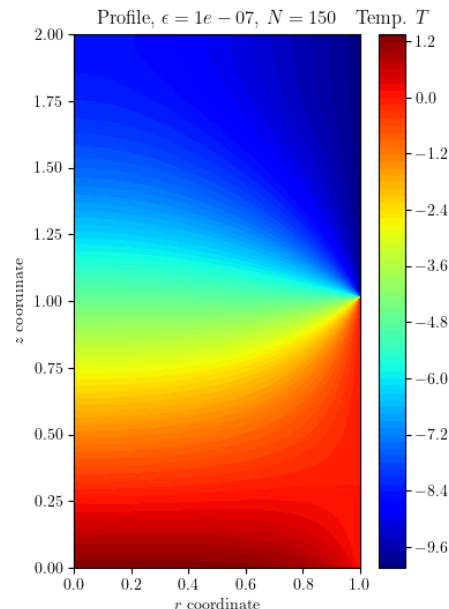


Figure 22: Temperature profile for $T_2 = -10$, $j = -10$ and an isolated top.

eprint: <https://doi.org/10.1080/00207160512331331129>. URL: <https://doi.org/10.1080/00207160512331331129>.

- [3] Zhiping Li. *Numerical Solutions to Partial Differential Equations*. [Last accessed: 24.03.2021]. URL: https://www.math.pku.edu.cn/teachers/lizp/courses/Numerical_PDE/Numerical%20PDE%20Lecture%20Slides/numpde_lecture_2_c1.pdf.

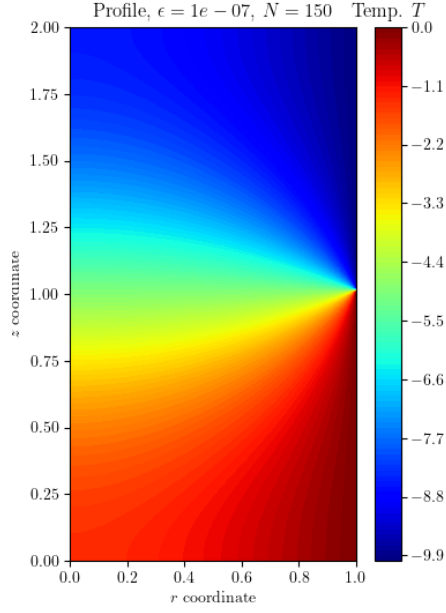


Figure 23: Temperature profile for $T_2 = -10$, $j = 0$ and an isolated top.

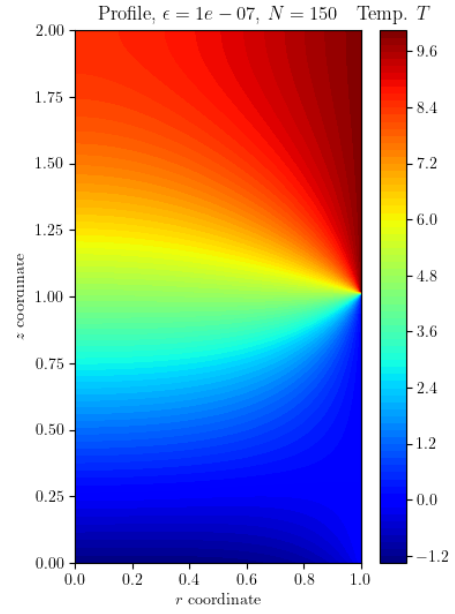


Figure 25: Temperature profile for $T_2 = 10$, $j = 10$ and an isolated top.

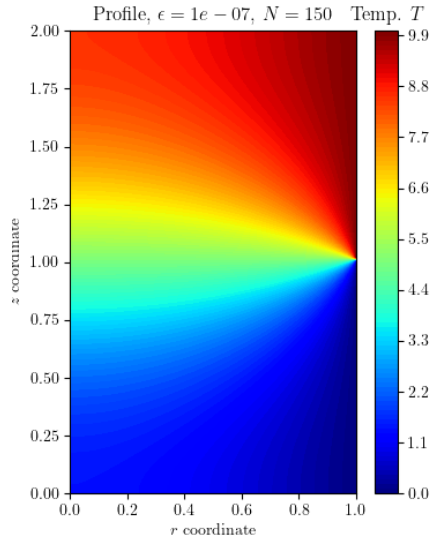


Figure 24: Temperature profile for $T_2 = 10$, $j = 0$ and an isolated top.