Direct Solving of the Poisson Equation

Aleksandar Ivanov

July 29, 2021

1 Problem Statements

Explore solutions of the following problems using Fast Fourier Transform-like methods (in our case sine transforms).

Problem 1

Find the sag at the center of a square membrane that's attached to a horizontal square frame due to its own weight. The mass distribution is as in task 6, and is also shown in fig. 1 where the density of the light part is normalized to 1, while the density of the darker part is $\tilde{\rho}$. You can try out the method on the homogeneous membrane where you can compare SOR, 1D FFT and 2D FFT.

Problem 2

We cool a metal cylinder of dimensions $H = 2R$ on both of its bases to a temperature T_1, while we heat its side to a constant temperature T_2. Find the temperature profile of the cylinder.

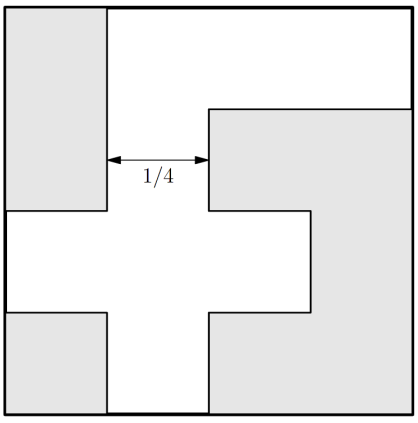

Figure 1: Mass distribution for the membrane.

2 Mathematical Setup

In dimensionless form, we're looking at solving the Poisson equation

$$\nabla^2 u = g. \tag{1}$$

In the first problem u represents the sag of the membrane, while g is a measure of its density from place to place. For the second problem, on the other hand, u represents the temperature, while $g = 0$.

For the square domain in Cartesian coordinates with Dirichlet boundary conditions, we know that sines of integer frequency are orthogonal functions, so it's a good idea to expand in terms of those, i.e. use the Discrete Sine Transform. We can also choose to use the Discrete Fourier Transform but in that case we would have to expand our domain on all sides by an odd mirroring, so that we get periodic boundary conditions.

Since we are working in two dimensions, with the boundary conditions clearly defining a preferred choice of axes, it's best to organize the solution into a matrix $u, g \in \mathbb{R}^{N\times M}$.

Using this and the convention that objects in Fourier space get upper indices we can write the discrete transform symbolically as

$$\hat{u}^{nm} = \sum_{ij} u_{ij} \sin\left(\frac{\pi i n}{N}\right) \sin\left(\frac{\pi j m}{M}\right), \tag{2}$$

and similarly for g. This, of course, we will not do by hand since the two sums for each element would give a time complexity of $\mathcal{O}(N^2M^2)$. Instead, we will use the fast sine transform which, similarly to the FFT, organizes the calculation in such a way as to get a time complexity of $\mathcal{O}\left(NM\log(NM)\right)$.

To solve the differential equation we need to transform it to Fourier space too. We can do this in two ways

1. first discretize and subsequently evaluate the finite difference approximation to the derivative on the already transformed $\hat{u}$ from eq. (2) or

2. apply the transformation to the continuous case, getting the wavenumber k_{nm} of the sines, and then discretize the objects.

Of course, in the limit of large N and M these two approaches have to agree. We will test the two against each other later.

Choosing the second option, the solution in Fourier space is

$$\hat{u}^{nm} = -\frac{\hat{g}^{nm}}{\pi^2(n^2+m^2)}, \tag{3}$$

where we have chosen to work on the domain with size 1×1.

Choosing option one, on the other hand, reproduces the formula given in the instructions

$$\hat{u}^{nm} = \hat{g}^{nm}\left(\frac{1/(2N^2)}{\cos(\frac{\pi n}{N})-1} + \frac{1/(2M^2)}{\cos(\frac{\pi m}{M})-1}\right). \tag{4}$$

In some situations, like in the case of problem 2, taking the FFT or the FST is not helpful in one of the directions because of the coordinate system or boundary conditions. In that case we can still profit from the speed of the FFT/FST by doing it only along one direction

$$\hat{u}^n{}_j = \sum_i u_{ij} \sin\left(\frac{\pi i n}{N}\right), \tag{5}$$

and similarly for g. The Laplacian then also transforms in the FFT direction, the same as discussed previously, but the other direction is still left with its ordinary space discretization leading to the familiar 1D tridiagonal system of equations, which has to be solved for each Fourier mode separately. Since this can be done in $\mathcal{O}(M)$, the whole procedure takes $O(MN \log N)$.

For the example of problem 2, the Laplacian is discretized into the system of equations.

$$\hat{u}^n{}_{i-1}\left(1-\frac{1}{2i}\right) + \hat{u}^n{}_{i+1}\left(1+\frac{1}{2i}\right) - \\ -\left(2+(k_z\Delta r)^2\right)\hat{u}^n{}_i = 0, \tag{6}$$

which hold for all interior points, where $k_z = \frac{n\pi}{2N}$, which depends on n so that the system is different for different modes.

For problem 1, the boundary conditions are Dirichlet on both sides, so they are already taken into account by simply using the DST. For Problem 2, on the other hand we have to work a bit more.

There we have the two parameters T_1 and T_2, however by redefining the zero and scale of temperature as $T \mapsto (T - T_1)/T_2$ we can get rid of both of them. We are then left with Dirichlet zero conditions in the z direction and a Dirichlet and Neumann condition in the r direction. The boundary conditions in the z direction are again taken into account by the DST, but for the r direction we have to apply them as we would when solving with the finite difference method. On the $r = R$ edge we have a Dirichlet condition which we implement by adding it as a source on the edge, as is usual. The important part here is that we have to perform the DST on the boundary condition column before adding it to the equation.

The application of the Neumann boundary condition is slightly more complicated. It is achieved by adding a phantom point on the outside of the boundary and requiring that $\hat{u}^n{}_{-1} = \hat{u}^n{}_1$, which changes the Laplacian discretization to

$$-\hat{u}^n{}_0\left(1+(k_z\Delta r)^2\right) + \hat{u}^n{}_1 = 0. \tag{7}$$

The second problem has inhomogeneous boundary conditions in the r direction, so it cannot easily be solved analytically. The homogeneous membrane from problem 1, however, can be solved analytically by doing the expansion in the sines, which gives us a solution as an infinite sum of weighted sines.

As discussed during lecture, choosing $g = 1$ gives the expansion coefficients

$$\hat{u}^{nm} = \frac{-4\left(1-(-1)^n\right)\left(1-(-1)^n\right)}{\pi^4 nm(n^2+m^2)}. \tag{8}$$

Using these and plugging in $(x, y) = (1/2, 1/2)$ for the point of maximum sagging, we get the value

$$u_{\max} = -0.0736713532..., \tag{9}$$

which we will compare against later.

3 Numerical details

For Python, 2D sine transform capabilities are provided by `scipy.fft.dstn`/`scipy.fft.idstn`, which will be the method which we use, while 1D sine transform capabilities are provided by `scipy.fft.dst`/`scipy.fft.idst`.

For solving the tridiagonal system of equations for the 1D FFT combined method, we will use `scipy.linalg.solve_banded`.

Timing of the processes was done with the Python module `timeit` and repeated with the module `time` to make sure of the results.

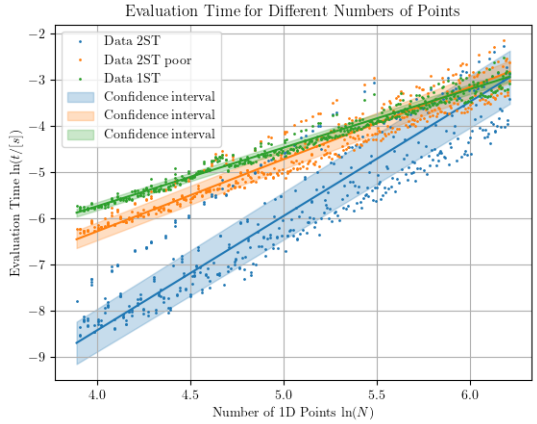

Figure 2: Fit for the evaluation times as a function of number of points $N = M$. Shown are the 2D built-in sine transform (blue), the 2 1D sine transform replacement (orange) and the combined 1D sine transform + matrix solving (green).

4 Results

The reason we're interested in these methods is their speed, so we start by testing that. We will compare the 2D sine transform against the two 1D sine transforms against the combined method. The results are shown in fig. 2 From it, we see that the 2D sine transform behaves as expected having a slope in the log-log graph of ~ 2. The two 1D transforms and the combined method present more surprising results. They have slopes smaller than ~ 2 (~ 1.5 and ~ 1.3) and would seem to say that we're somehow computing the whole procedure asymptotically faster than we would with the 2D sine transform. This result seems impossible since it would mean that we're computing the Fourier transform faster than $\mathcal{O}(N \log N)$. Timing was done with both the `timeit` and `time` modules for Python; both gave similar results. External timing with the bash `/usr/bin/time` command gave times that didn't support the preceding, however because of Python's large overhead, the results are inconclusive. In all cases processing was done on only one core.

The timing method is not even consistent with itself since if the 2D FFT is slower we would expect it to overtake the other two after some number of points which doesn't happen. Namely, testing at $N = 9000$, which would be well into the region where two 1DFFTs dominate according to fig. 2, we get that the evaluation times are comparable for both the 2DFFT and two 1DFFTs in the sense that which of the two wins is random.

I have not been able to figure out what is causing this effect, which is presumably just an artifact. Outliers definitely play a role in artificially increasing the fit value for the 2DFFT as can be seen from fig. 2, but the other two seem to be well approximated by the fit used, and they still show abnormally small values.

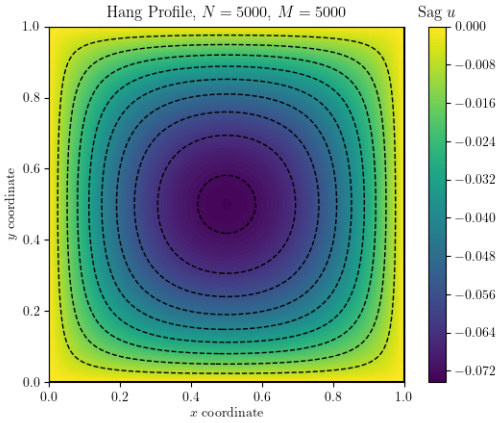

Figure 3: Profile of the homogeneous membrane with $N = M = 5000$.

Having considered the homogeneous membrane analytically, we turn to solving it numerically. Using $N = M = 5000$ we calculate the profile shown

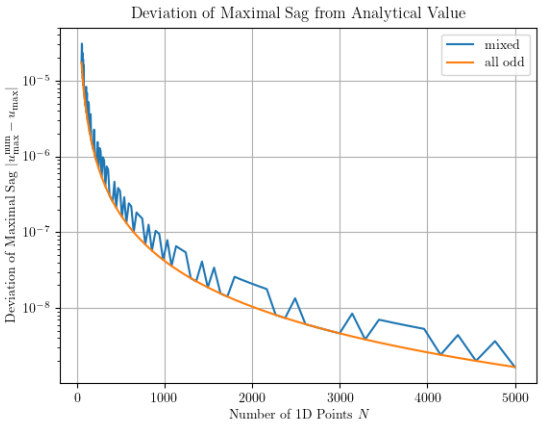

Figure 4: Convergence for the homogeneous membrane.

in fig. 3. At this number of points the numerically calculated maximal sag is

$$u_{\max}^{\text{num}} = -0.073671349948, \tag{10}$$

which, comparing with the analytical result is correct to eight decimal places.

To better see how this value converges we plot the maximal sag as a function of the number of points that it has been calculated with. Figure 4 shows exactly this by plotting the deviation from the analytical value. We see that the result converges very fast at the beginning but slows down afterwards. Another interesting features is that the convergence is not monotonic but rather seems to jumps between two different curves. This is explained by the orange curve as it is just an artifact of the fact that if we work with an even number of points then we don't have a point in the middle where the maximal displacement is. Generically, we can't expect to have a grid point exactly at the minimum, but this graph assures us that even if we are off by a half a grid spacing, which is the maximum we can be, the results are not too far off. Another form of non-monotonicity that can't be seen in the log plot is that the values calculated numerically are sometimes larger and sometimes smaller than the analytical value, but still overall converging to it.

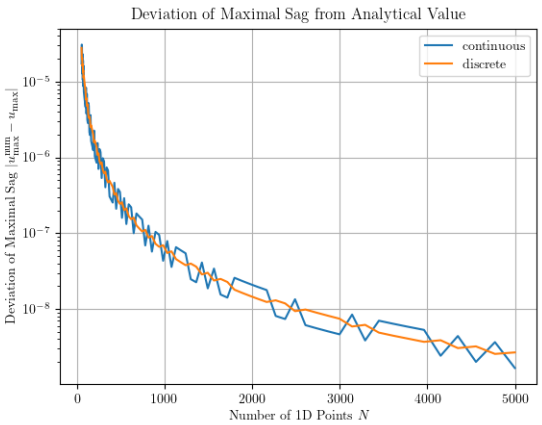

Figure 5: Error for the two orders of discretization mentioned before. The curve 'continuous' is the order where the discretization is taken after(eq. (3)), while 'discrete' is the order where discretization is done before(eq. (4))

Figure 5 shows the results of what happens when we switch the steps of discretization and transformation. We see that the line marked 'continuous', which represents transforming first has more variance between whether we have an even or odd number of points but the average value tracks the line 'marked' discrete, which represents discretizing first. This means that both give similar accuracy but discretizing first is more stable, in a sense.

With all this technology we can finally solve the

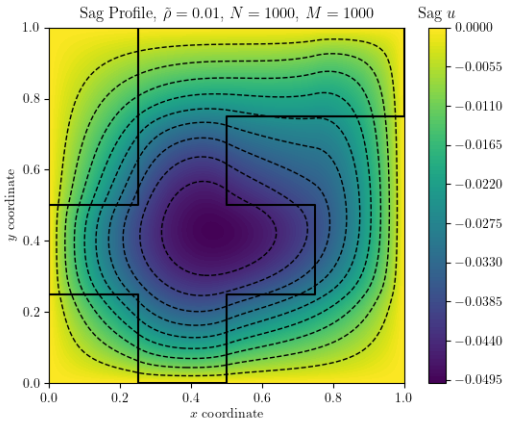

Figure 6: Sag profile for the F-shaped distribution with $\tilde{\rho} = 0.01$

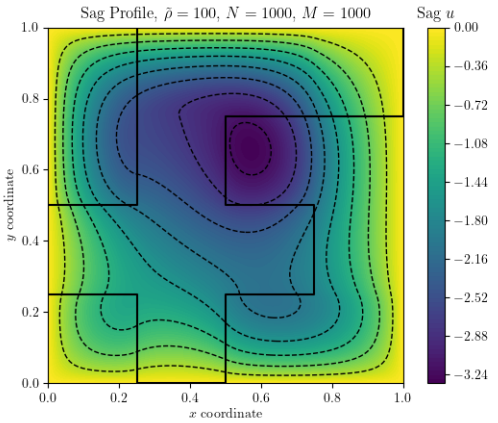

Figure 7: Sag profile for the F-shaped distribution with $\tilde{\rho} = 100$

physical problem — what is the shape of the membrane for the F-shaped distribution. Of the methods above, we choose the 2D FFT one because both directions support it. Doing the calculation with $N = M = 1000$ and the densities $\tilde{\rho} = 0.01$, $\tilde{\rho} = 1$ and $\tilde{\rho} = 100$, we get figs. 6 and 7 They show that the profile looks very different depending on whether $\tilde{\rho}$ is smaller or larger than 1. For $\tilde{\rho} = 1$ we, of course, recover the solution for the homogeneous membrane fig. 3. For small densities the maximum sag lies within the F-shape, while for large densities it lies outside of it.

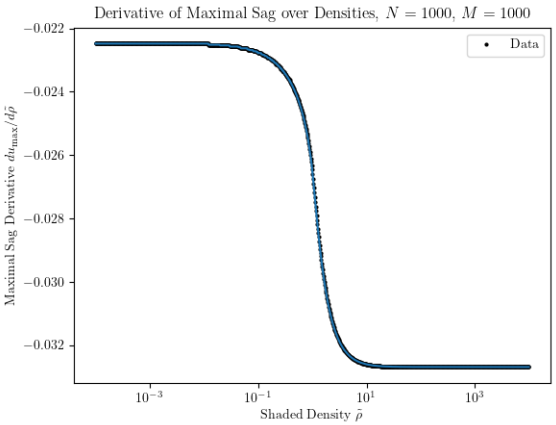

Figure 8: Derivative of maximal sag w.r.t. $\tilde{\rho}$ as a function of $\tilde{\rho}$. Density parameter is on a log scale.

We're also interested in the maximal sag as a function of the density. In the case of the homogeneous membrane this is almost trivial since the equation is linear the maximal sag will be a linear function of the density. In the case of the F shaped membrane, on the other hand, this need not be the case since the density we're changing $\tilde{\rho}$ changes g in a nontrivial way. fig. 8 shows the results of calculating the derivative of $u_{\max}$ with respect to $\tilde{\rho}$. We do this because for large values of $\tilde{\rho}$ the dependence is linear. This result is expected since for those values 1 is negligible compared to $\tilde{\rho}$ and the result just scales linearly as in the homogeneous case. What we learn from this graph is that for small $\tilde{\rho}$ the dependence also scales linearly and in between, around $\tilde{\rho} = 1$, there is a transition period. The jumps are as before artifacts of how close a grid point is to the actual maximum. Figure 9 shows the same data as in fig. 8 but now without taking a derivative and on a log-log scale. We can see the two regions that fig. 8 shows.

Another interesting quantity to look at is the position of the maximum sag as a function of the density $\tilde{\rho}$. This is because the $\tilde{\rho} < 1$ and $\tilde{\rho} > 1$ profiles look very different from each other. Following the density from 10^{-4} to 10^{4} with a logarithmic distribution of points between (a geometric series of points), we generate fig. 10 it shows how the posi-

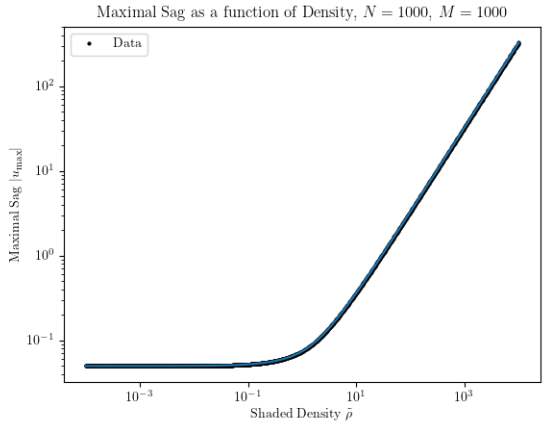

Figure 9: Maximal sag as a function of $\tilde{\rho}$ plotted in a log-log scale. The two slopes fig. 8 implies is again seen, although the smaller slope is too gradual to be seen clearly.

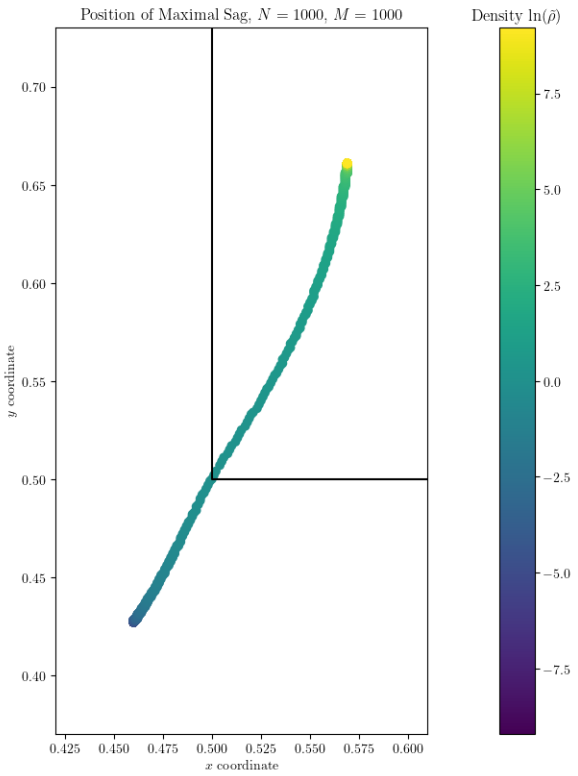

Figure 10: Position of maximal sag as a function of $\tilde{\rho}$. Density is sampled from a logarithmic distribution.

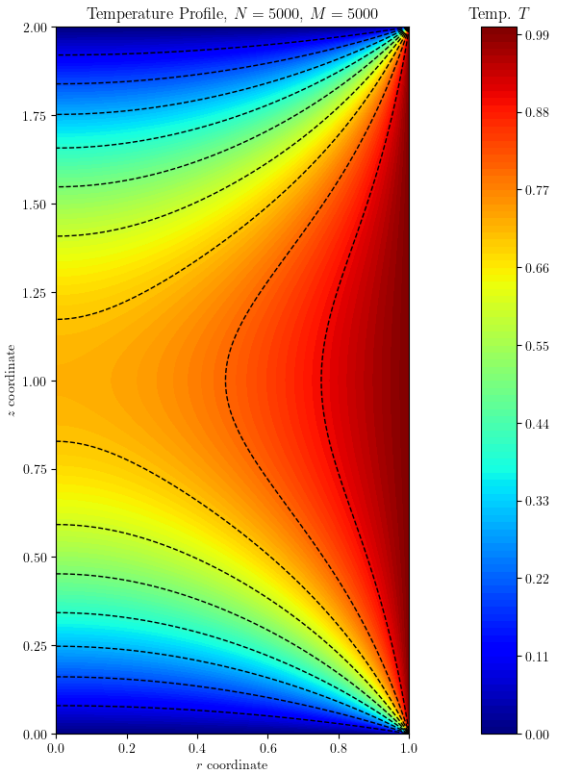

Figure 11: Temperature profile for the cylinder.

tion of the maximum moves with density. We see that this position reaches some limiting points in the $\tilde{\rho} \to 0$ and $\tilde{\rho} \to \infty$ limits and moves between the two for values in between. From the relatively uniform color of the line we see that the maximum moves across and settles in one of the two limiting points relatively fast. This is the same conclusion we reached from fig. 8, where the transition period happens fast.

For the second problem as already talked about in the setup of the problem, if we eliminate the extraneous parameters, we see that the problem doesn't have any free parameters, so there is only a single solution. The combined method that we described before is convenient for this problem and using it we get the temperature profile shown in fig. 11. We see an expected sight; the temperature is constant along the edges, the radial flow is 0 on the z-axis, and we have symmetry along the middle of the cylinder.

If we want to continue using the DST in the vertical direction means that we should stick to Dirichlet boundary conditions on the top and bottom. The radial direction's boundary conditions, on the other hand, are more tuneable. We are forced to keep the $\partial T/\partial r = 0$ Neumann condition at $r = 0$ since this is forced onto us by the geometry, but we can change the boundary condition on the outside to any function we want.

The first thing we try is the sine function, i.e. we set the boundary condition as

$$T(R, z) = \sin(\pi z). \tag{11}$$

This generates fig. 12, where we can see that the overall temperature profile is much more moderate when compared to fig. 11. This fact makes sense, since if we're putting in a sine boundary condition then after the heat has spread around we expect to have put in less energy overall — the negative and

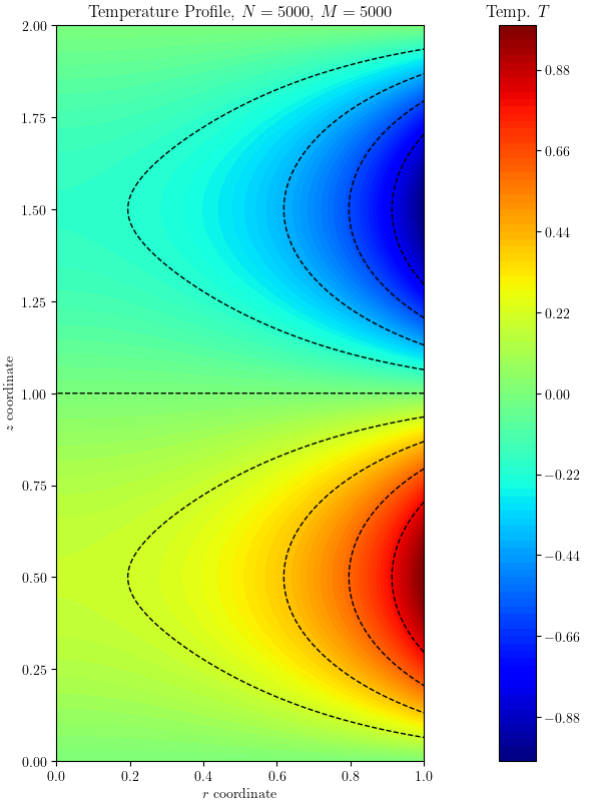

Figure 12: Temperature profile for the cylinder given a sine boundary condition.

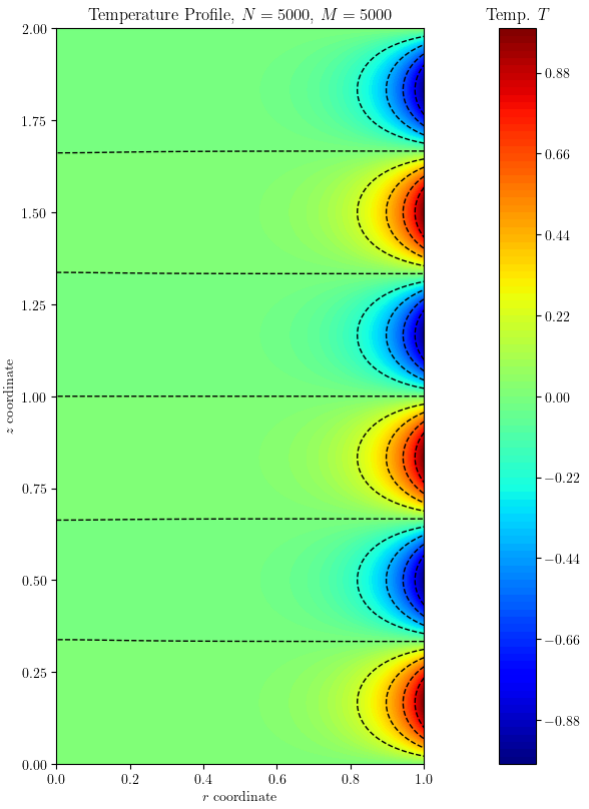

Figure 13: Temperature profile for the cylinder given a sine boundary condition with a larger frequency.

positive parts of the sine contribute in opposing ways. Figure 13 corroborates this since it shows that if we increase the frequency of the sine the then the eventual heat profile is even more moderate. In both cases, though, there is some amount of variation along the inner $r = 0$ boundary, which is shown by the straight contour lines being in the picture.

Another thing to notice is that now there is no bunching up of the contour lines in the right-most corners. This is because our boundary conditions are now compatible in the corners. Namely, the sine boundary condition and Dirichlet boundary conditions agree that the temperature in both corners is 0. This wasn't the case with the homogeneous boundary condition which said that the temperature was 1 in the corner, which is why the contour lines have to very quickly transition from 0 to 1, and they bunch up.

Figures 14 to 16 show some other possible configurations for the boundary condition. In fig. 14 we have only a small region where the temperature is non-zero, and, as before, this small amount of heat means that only a very small region is above $T = 0$. Figure 15 has a normal distribution of temperatures, while fig. 16 has two of them, but centered on the corners instead of the middle. In a similar vein, fig. 17 shows an asymmetric boundary condition that is only one side of the distribution in fig. 16.

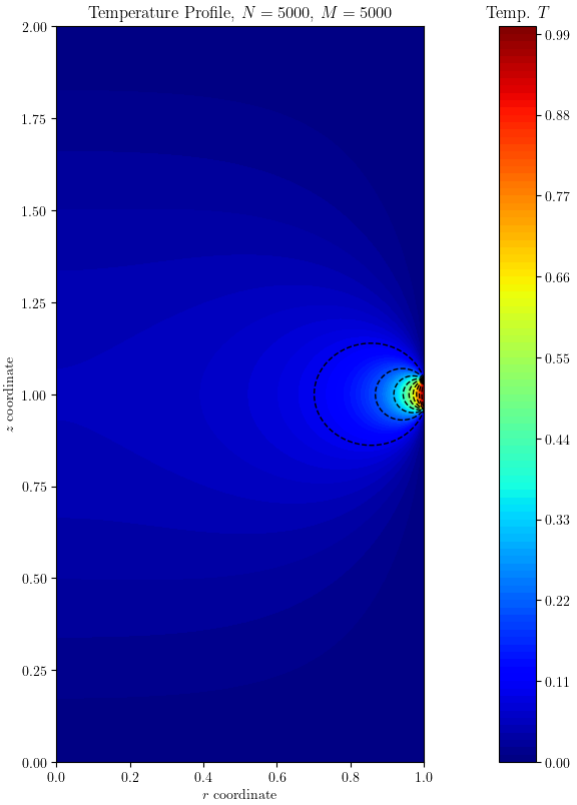

Figure 14: Temperature profile for the cylinder given a boundary condition that is only non-zero in a small region near the center.

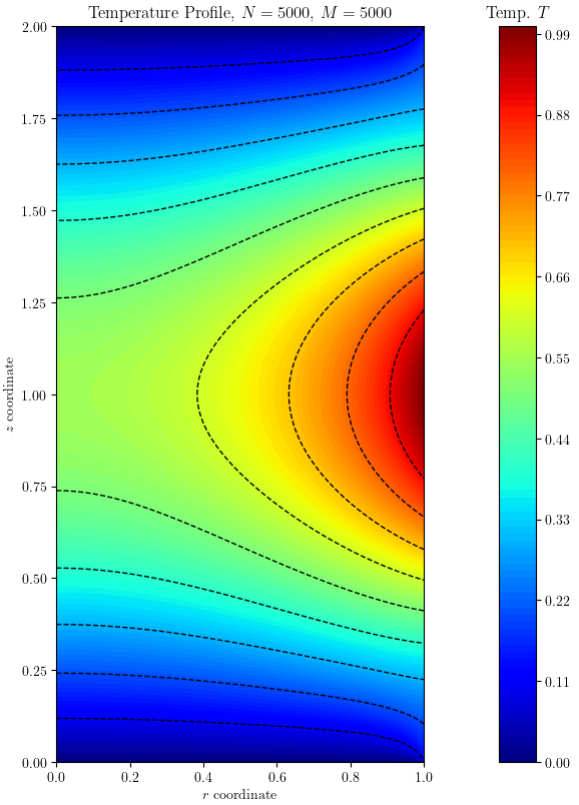

Figure 15: Temperature profile for the cylinder given a normally distributed temperature on the boundary.

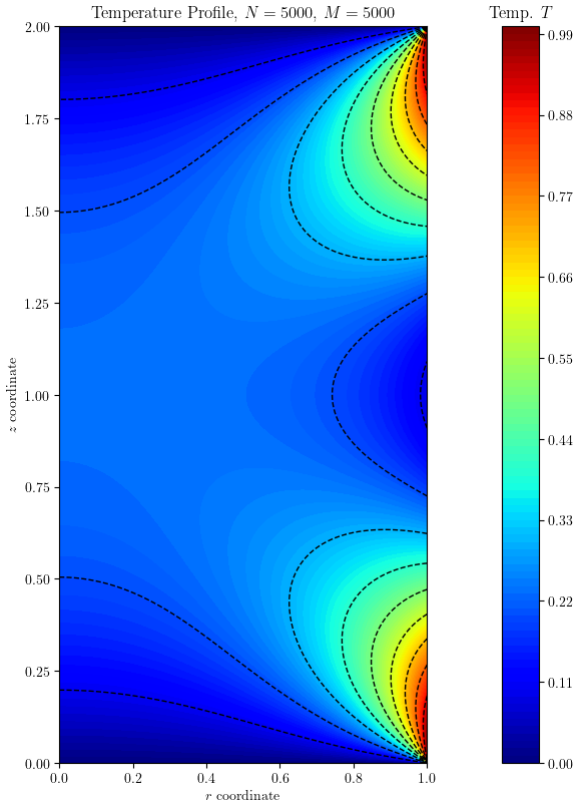

Figure 16: Temperature profile for the cylinder given boundary condition that is off-center.

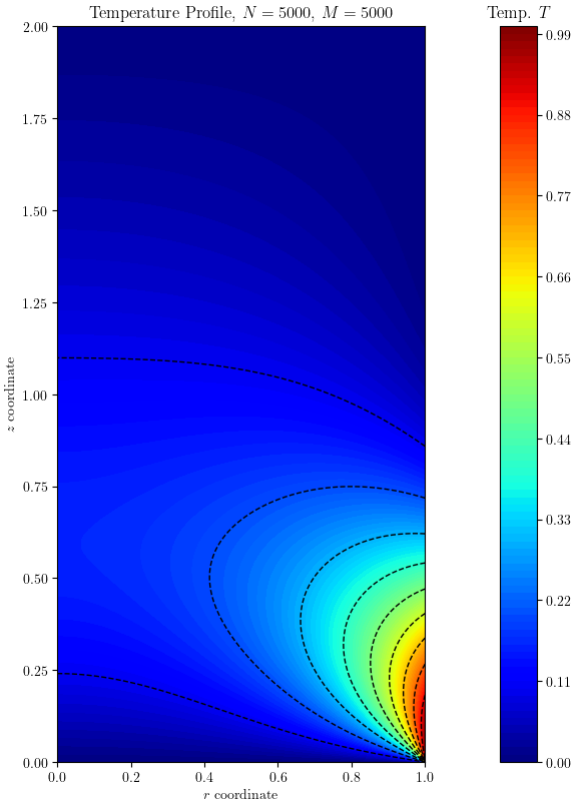

Figure 17: Temperature profile for the cylinder given boundary condition that is asymmetric.

We don't have to stick to adding sources only on the boundary, we can also and sources and sinks on the inside. This is done by changing the Laplace equation into a Poisson equation, which is not much work for us with the methods we're using.

Figures 18 and 19 show this for a normally distributed source that is the equivalent of a ring in the 3D geometry, and a source that is only non-zero in a very small region, respectively. The boundary condition in both cases is

$$T(R, z) = 0. \tag{12}$$

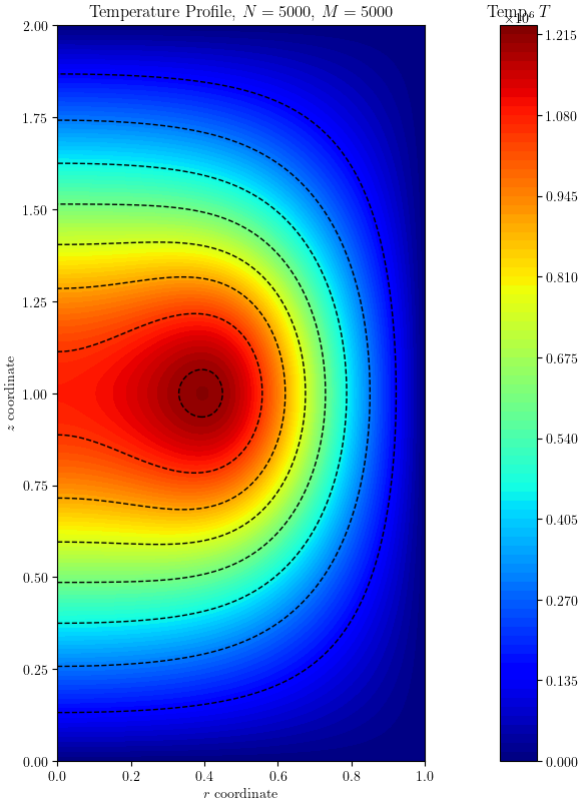

Figure 18: Temperature profile for the cylinder with a normally distributed ring source at $r = R/2$, $z = H/2$.

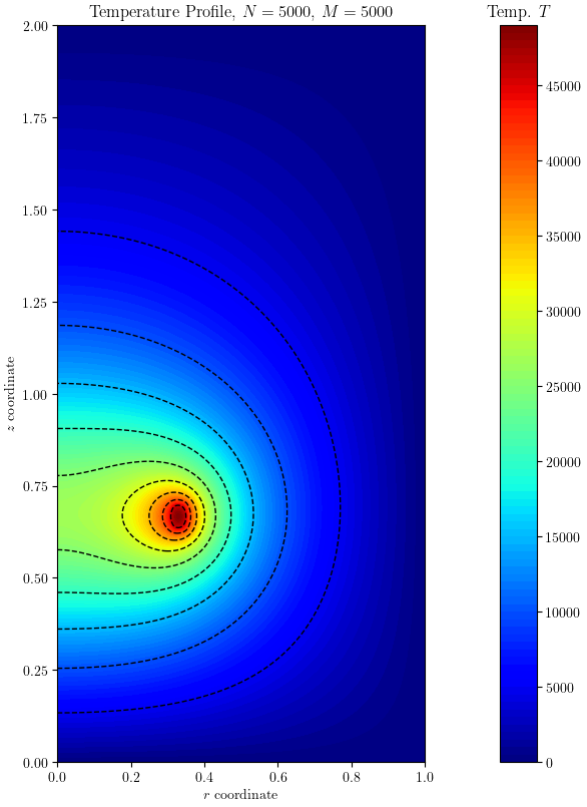

Figure 19: Temperature profile for the cylinder with a ring source of square cross-section in an asymmetric position.