**2a**

Method used: Gauss-Seidel

100 grids, 10,000 iterations

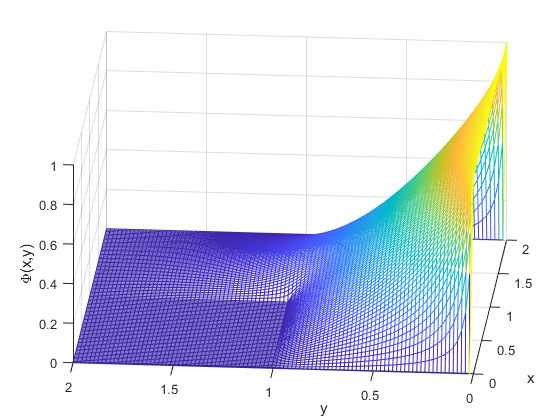
To deal with the irregular domain, I added a fictitious square to complete the domain into a square. Within this arbitrary domain, I set the value of phi to be 0 at all times. In the code, this translates to:

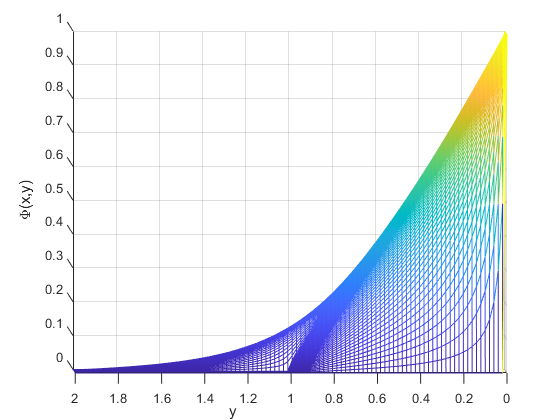
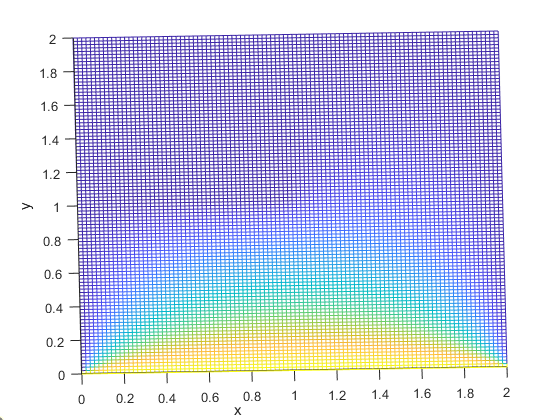
phi(1:N/2,(N/2+1):N) = 0

I modified the initialization condition given in Garcia’s code to make phi0 = 0, and incorporate the perturbation via the boundary condition [phi(:,0) = 1].

Since this is an elliptic equation, the time step is irrelevant, and we simply need the simulation to run long enough to reach steady state conditions. Theoretically, we’ve chosen the largest delta-t possible to make the stability parameter 0.25. However, the simulation is set to stop after N2 iterations which was the case here.

The results are shown below.





**2b**

I modified the code to incorporate the point charge at (0,0.5). since the length of each side is 2, this point can be written as (1,N/4). This update is ensured in every iteration in the code, giving the response below.

This approximation basically assigns the value of the point charge to a specific grid, not a specific point in space as was specified. Thus, for this to be more accurate, finer grid resolution would be beneficial.

