

1.

a) The first example requires 495 iterations to reach the desired tolerance while the second example only requires 9 iterations.

b) The matrix appears to be divergent between when  $w > 0.35$  (and is within the valid range for  $w$ :  $0 < w < 2$ ).

When  $0 < w < 0.35$ , it converges. The number of iterations seem to fluctuate in a way in which there is not a predictable minimum iterations.

This is expected because the smaller that  $w$  is, the less chance that overshoot as it was doing when  $w=1$ .

2.

a)

Second Order Linear ODE

$$\phi(x) = c_2 x + c_1 + 4\pi \sin(x),$$

we can make the constants = 0 since  $\phi(0) = 0$

$$\phi(x) = 4\pi \sin(x),$$

b)

finite-difference structure:

100x100 matrix of :

$$A = \begin{bmatrix} -2 & 1 & 0 & \dots & 0 & 1 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ 0 & \dots & 0 & 1 & -2 & 1 \\ 1 & 0 & \dots & 0 & 1 & -2 \end{bmatrix}$$

There is a 1 in the top right and bottom left corner in order to satisfy the periodic boundary conditions.

$$b = -4\pi \sin(x_i), \quad 100 \times 1$$

$$Ax = b, \text{ solve for } x$$

c) In the code.

For some reason I was unable to get myJacobi to work well on this matrix.

When I ran the function myJacobi it would run for a very long time, too long for me to wait.

However, if I waited long enough it would work. (~10 minutes)

During this comparison I saw that they followed similar patterns, but the

analytical solution was much more smooth.

I also compared the analytical solution to the numerical solution provided by numpy.

This is shown in my code and does not make much sense to me, the numpy solution seems to be magnitudes smaller than the analytical solution.

If you would like to see the comparison to my numerical solution you may uncomment lines 91

```
91: #x_numerical = myJacobi(A, b , 1., 10**-4)
```

and line 108

```
108: #plt.plot(x_analytical, x_numerical)  #UNCOMMENT THIS TO  
TEST MY NUMERICAL SOLUTION
```

3.

a)  $Ax = b$

A is matrix of  $d^2T/dx^2 + d^2T/dy^2$ .

b is the temperature vector.

x is the final temperature we solve for.

b) The equation of row 6 of the matrix is

$$T_3 + T_5 - 4T_6 + T_9 = 100$$

If you add up the matrices  $d^2T/dx^2$  and  $d^2T/dy^2$ , you would get this matrix

c)

The temperature values do not make sense given the boundary values of the temperature.

However if you multiply the A matrix by -1, we get values that make much more sense. (Shown in the code).

$$-Ax = b$$

I still get that the value of  $T_1$  is negative, but that is okay because it is the furthest away from the heat source.