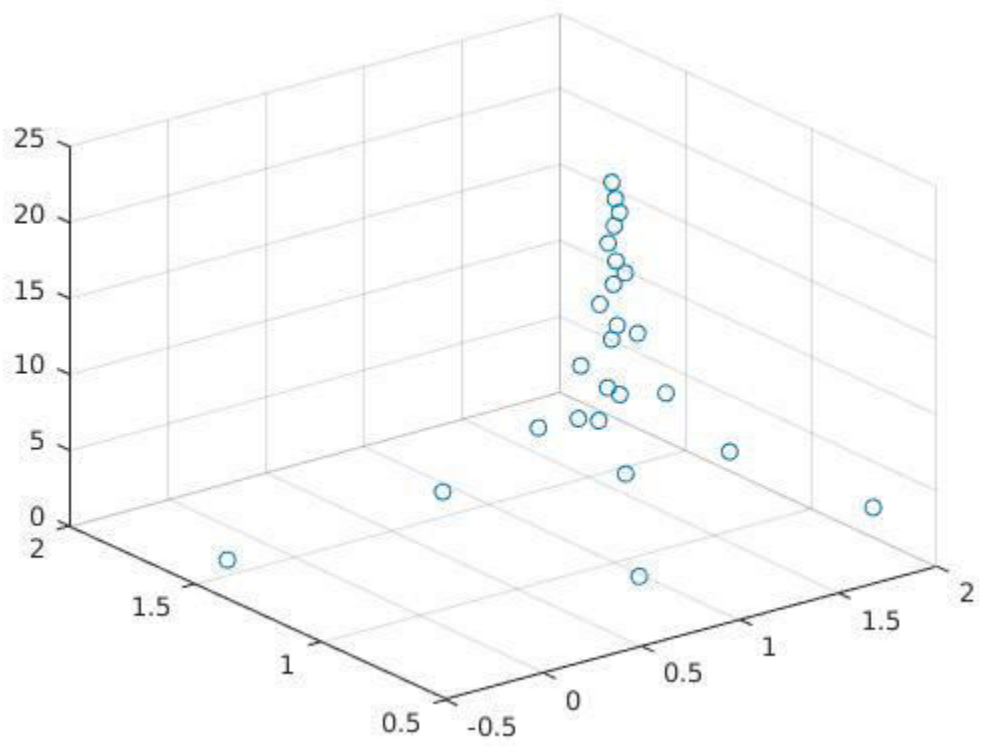
1.

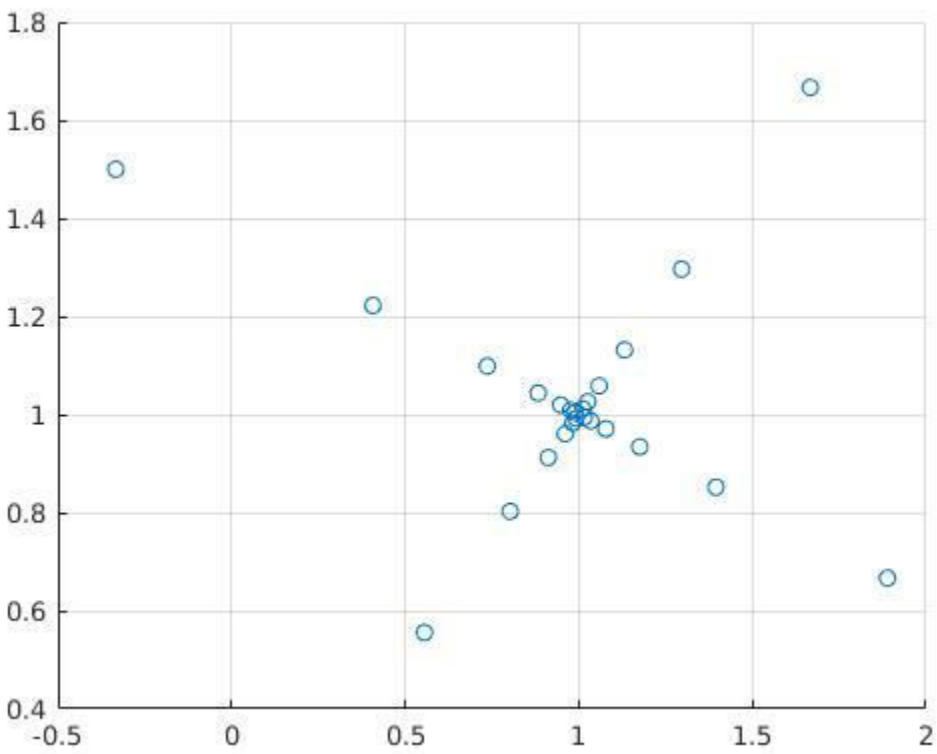
a. $x_{i+1} = x_i - (1/3)(3x_i - 4y_i + 1)$
$y_{i+1} = y_i - (1/2)(x_i + 2y_i - 3)$

b.

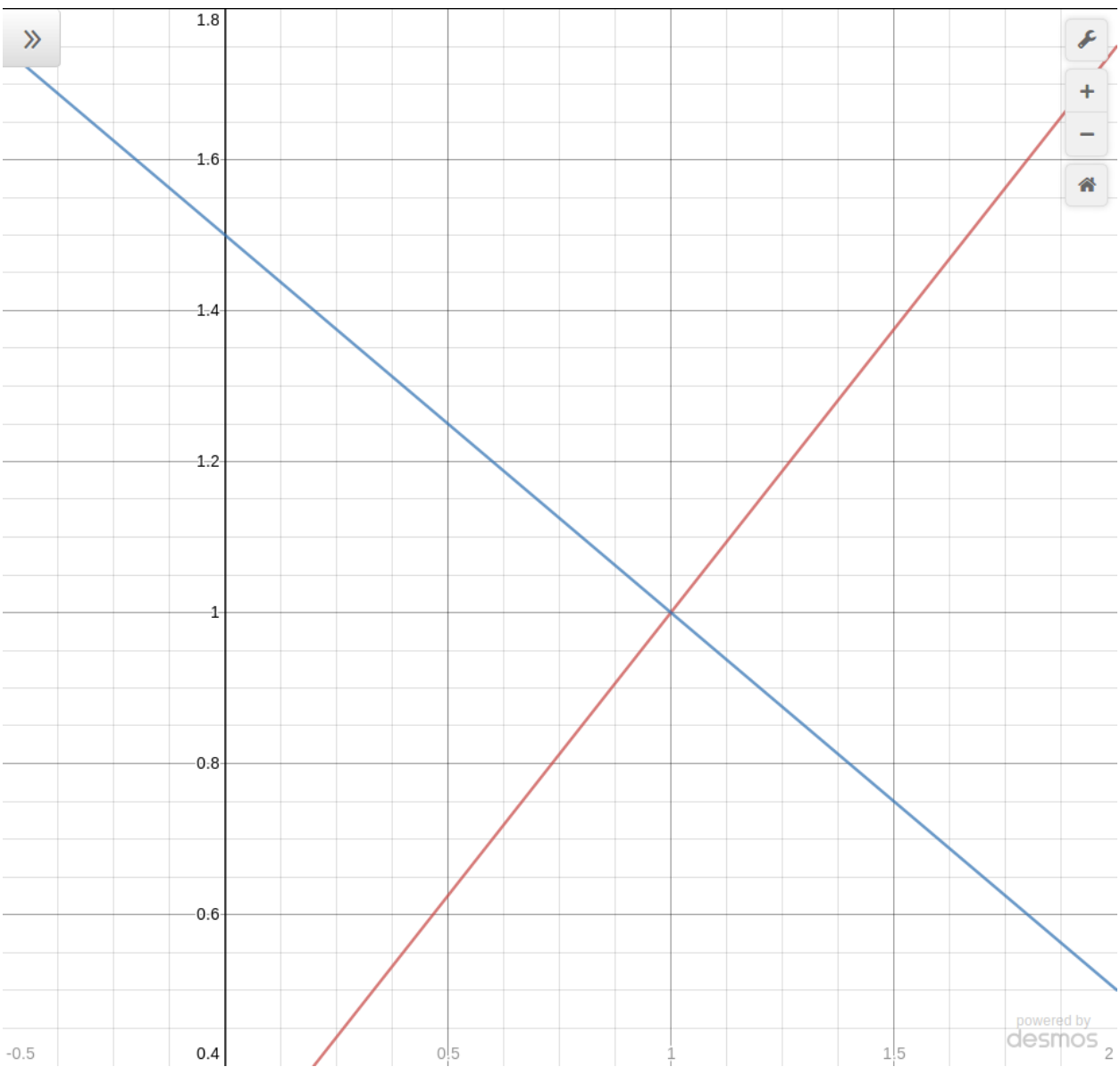


As the z-axis increases, the (x, y) pairs converge to the true values (1, 1) of the solution of the system. When the number of iterations are low, the residual error is high, because of the lack of previous computations to base the current solution off of. To get closer to the true value of the solution of the system, more iterations can be done.

Top-down perspective:



2-D graph of the linear system:

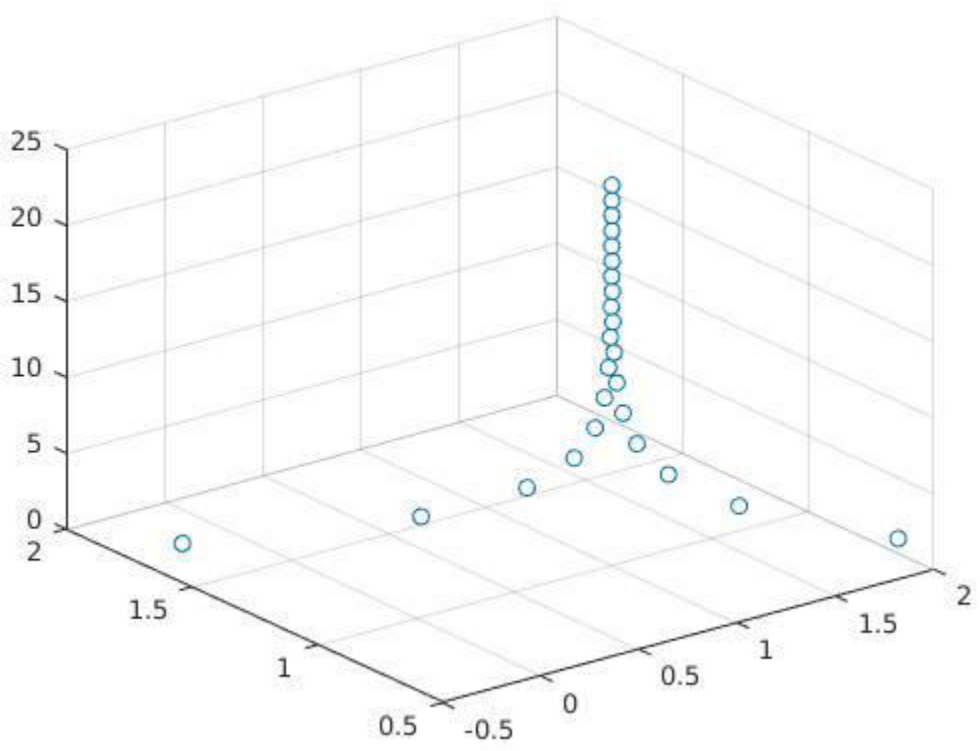


When looking at the plot from a top-down view, the (x,y) pairs fall along the lines of the original linear system. This is because the solution to each iteration of the recurrence relation is a function of some input (x, y) to one of the equations in the linear system. As the number of iterations increases, the points on the 3D plot converge onto the point where the lines meet.

c.

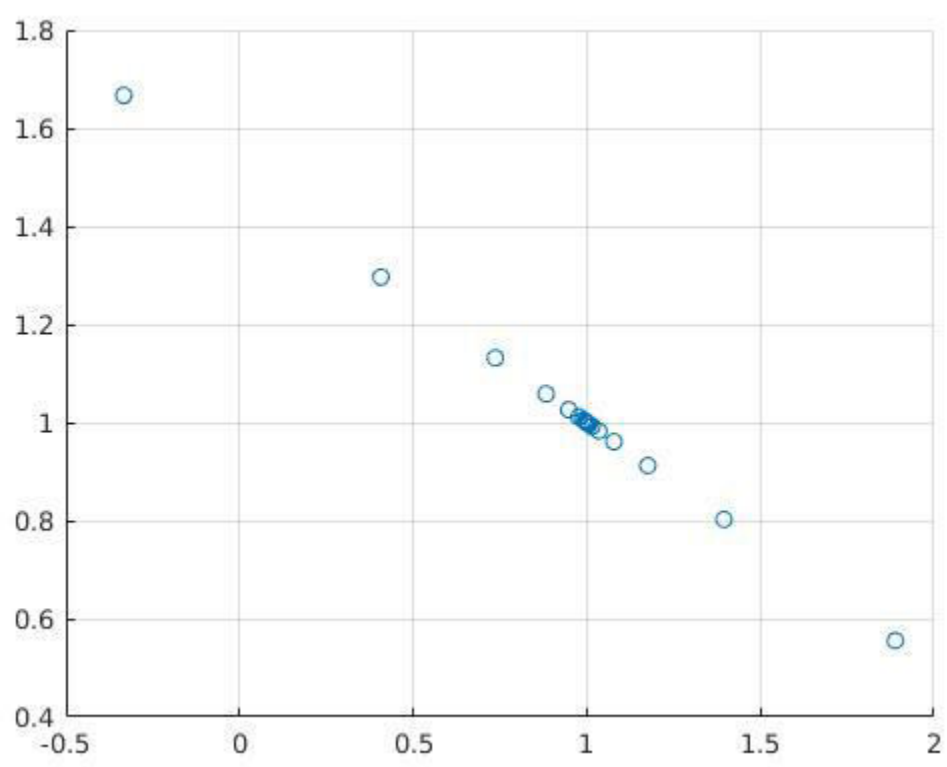
a. $x_{i+1} = (-1 + 4y_i)/3$
$y_{i+1} = (3 - x_{i+1})/2$

b.



As with the jacobi method, the (x, y) values of the iterative solution converge to the solution of the linear system. When the number of iterations are low, the error is high, so the (x, y) values are far from the solution of the linear system, but become more accurate as the number of iterations increases. The values seem to converge faster than the Jacobi method. This could be due to the fact that the Gauss-Seidel method uses the calculated solution values as soon as they're available when estimating the second equation, instead of waiting until the next iteration like the Jacobi method does.

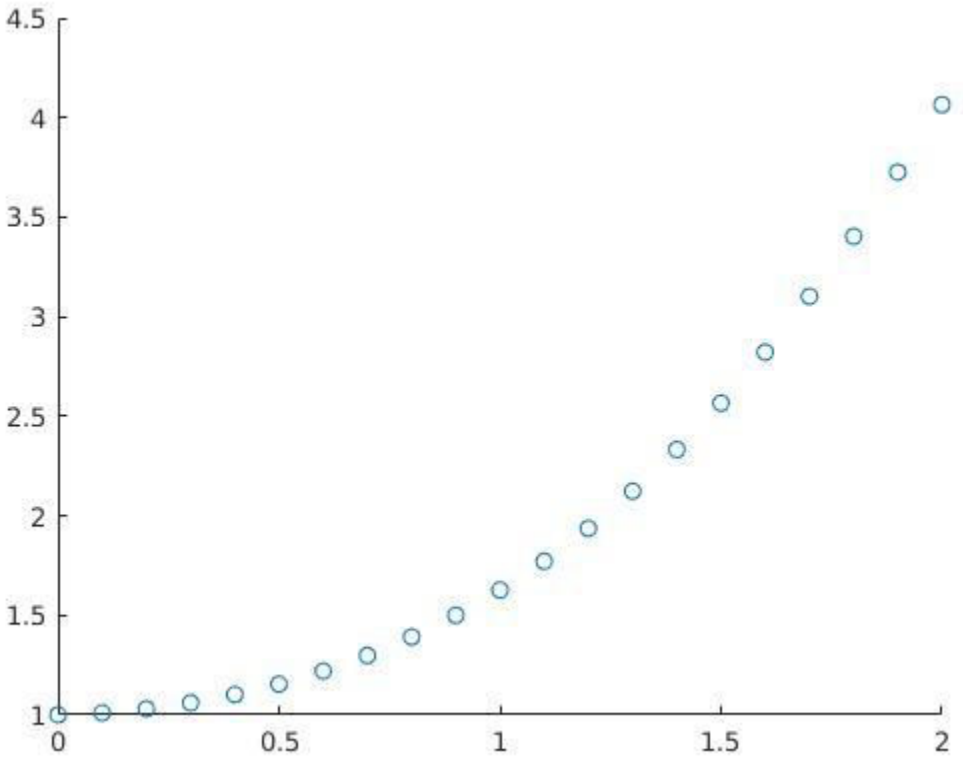
2D top-down view:



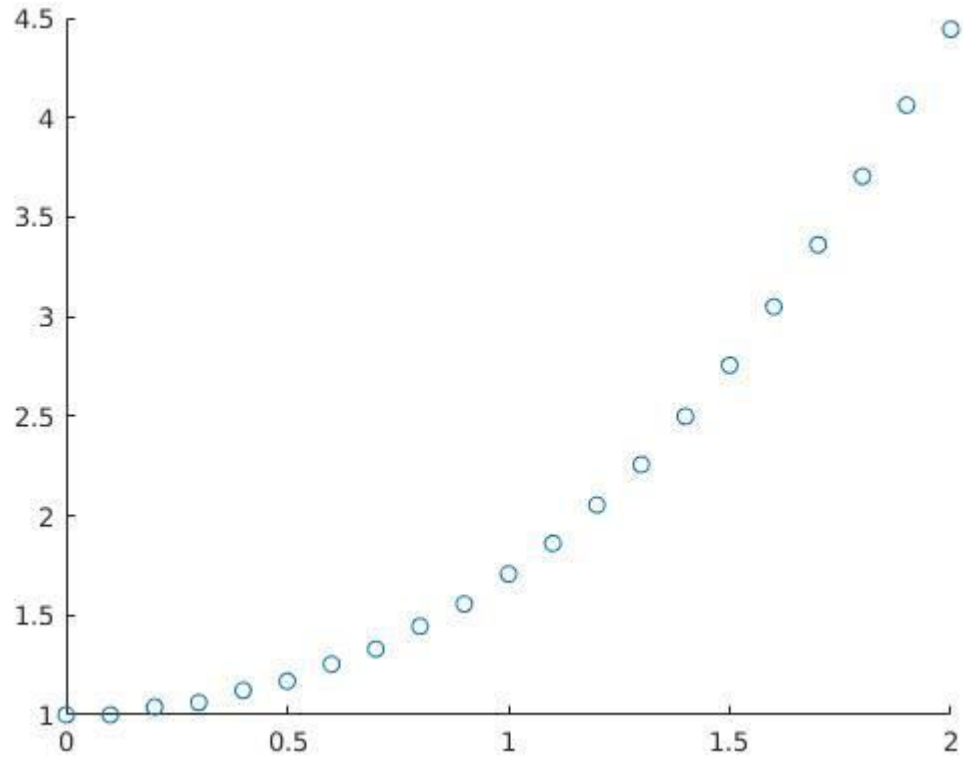
Unlike the Jacobi method, the (x,y) plots of the iterations of the Gauss-Seidel method only fall along the second linear equation in the system. This is because, unlike the Jacobi method, the recurrence relation doesn't depend on x_i to estimate the next iteration, so all of the points end up falling somewhere along the line of the second equation.

2.

a. $y_f((i+1) * h) = y * h * \sin(i * h) + y_f(i * h)$

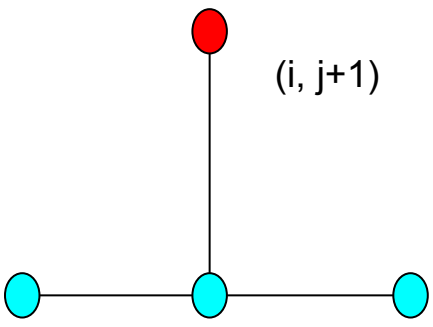


b. $y_c((i+1) * h) = 2h * y * \sin(i * h) + y_c((i - 1) * h)$



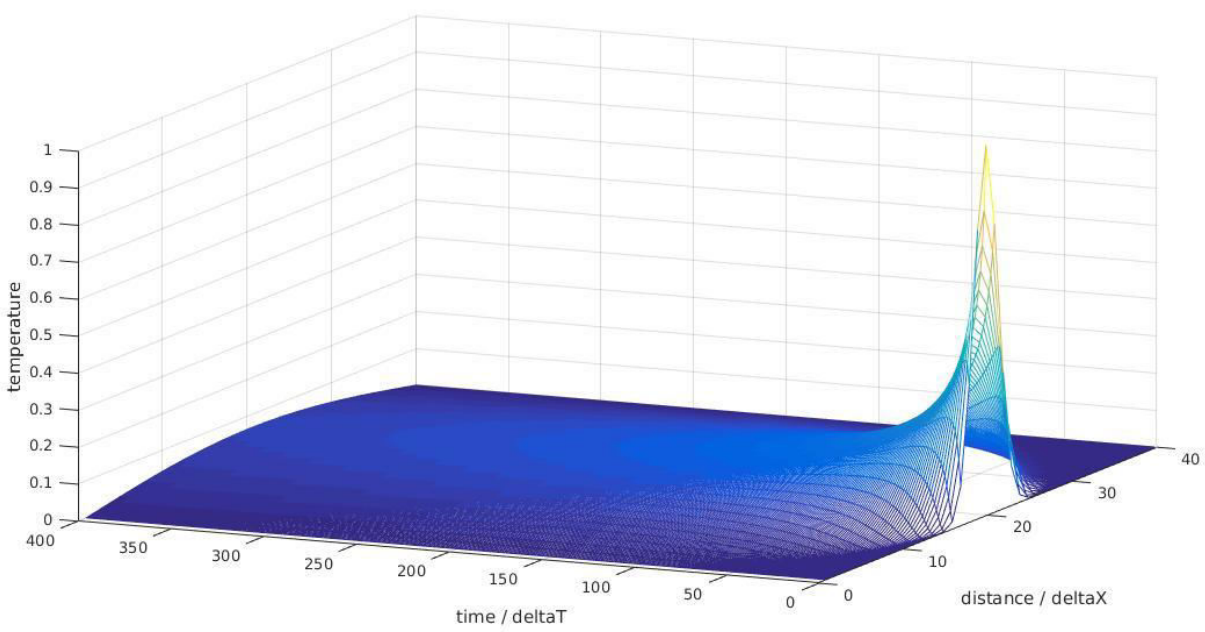
3.

a.



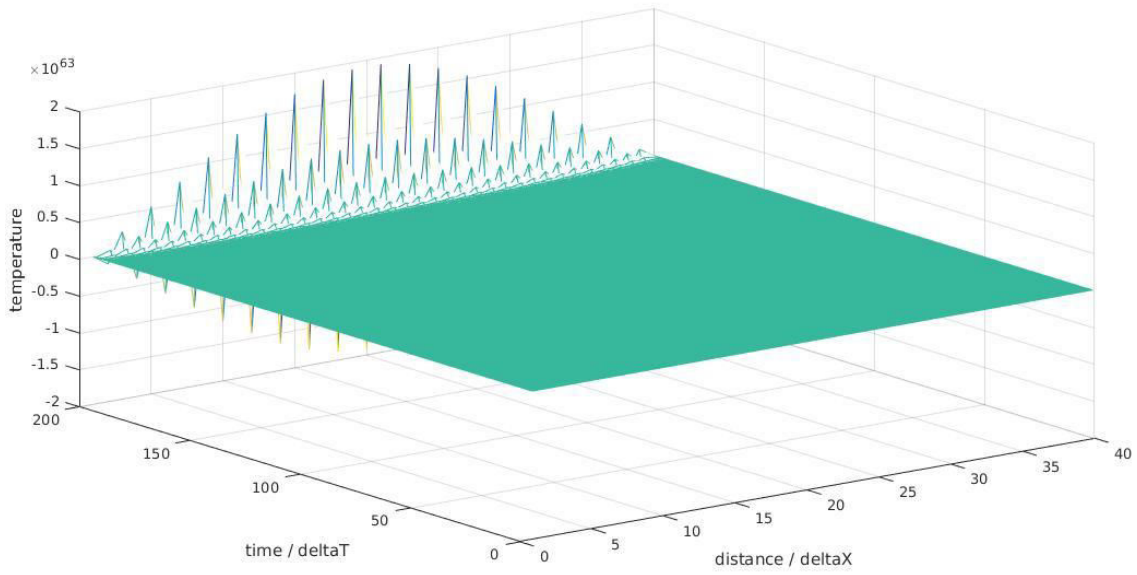
(i-1, j) (i, j) (i+1, j)

b.



This graph jives with our intuition, because as the time increases, the temperature dissipates over the length of the rod. When time / deltaT = 0, the temperature at the center of the rod is equal to 1 and drops off sharply to either side, but as the time increases, the temperature at the center of the rod starts to fall and the temperature away from the center increases slightly. As time passes, heat also dissipates out of the ends of the rod, so the total temperature of the rod should be decreasing, shown by a general decrease in temperature along all points as time increases.

c.



This plot does not jive with our intuition, because, as time increases, the temperature fluctuates wildly between $2*10^{63}$ and $-2*10^{63}$, which is impossible, since the maximum temperature when time equals zero is one, and negative temperatures are impossible. This is because, when deltaT is too large, then the constant (D * deltaT / deltaX^2) isn't small enough to prevent rapid exponential growth, as can be seen by the recurrence relation:

f(i, j+1) = (d * deltaT / deltaX^2) * (f(i-1, j) - 2 * f(i, j) + f(i+1, j)) + f(i, j)

Since each subsequent timestep depends on the previous timestep, then eventually the value of (D * deltaT / deltaX^2) * -2 * f(i, j) + f(i, j) becomes exceedingly large and fluctuates between negative and positive.