UNIVERSITY OF CALIFORNIA, DAVIS

MAT 128C - NUMERICAL ANALYSIS

PROFESSOR J. DE LOERA

Final Project

Ali Heydari
IN COLLABORATION WITH ALINA GATAULLINA, PROFESSOR DE LOERA AND STUDENTS FROM THE OFFICE HOURS

May 22, 2018

**** To run all the provided code, please press "run" in MATLAB and follow the prompt ****

Computer Project 1

In this problem, we are trying to experiment with the differential equations and values to explore different phenomenas from this model.

- Part I:
  For the first part, we have $W = 80mph$ (the wind speed); in this scenario the bridge is NOT stable. The small disturbances, θ do not die out and they keep growing as the time goes on, which makes the bridge unstable in periodic motion, and the time goes to infinity, the behavior becomes chaotic as seen in the animation produced by the corresponding code. (Corresponding code: tacoma_1.m)

- Part II:
  In the second part, we aim to improve the approximation from part one by implementing the fourth-order Runge-Kutta method; in addition, we plot the $y(t)$ and $\theta(t)$ versus time. In the approximation, we used the text book equations for Runge-Kutta:

$$
\begin{aligned}
k_1 &= \dot{y}(t, w) \\
k_2 &= \dot{y}(t + \frac{h}{2}, w + h \cdot \frac{k_1}{2}) \\
k_3 &= \dot{y}(t + \frac{h}{2}, w + h \cdot \frac{k_2}{2}) \\
k_4 &= \dot{y}(t + h, w + h \cdot k_3)
\end{aligned}
$$

  Using Runga-Kutta increased the accuracy but not so much the efficiency (compile time) of the code. (Corresponding code: tacoma_part2.m).

  Now, when we graph the two function $y(t)$ and $\theta(t)$ we find that in figure 1 $y(t)$ vs. *time*, the oscillations decay to a threshold, which indicates periodic behavior. The θ vs. *time* graph shows oscillations of the angle will also oscillate withing a certain threshold, but the disturbances do not die out. This makes sense since the wind is constantly blowing, therefore making the bridge have a pendulum-like motion after a certain t_i.

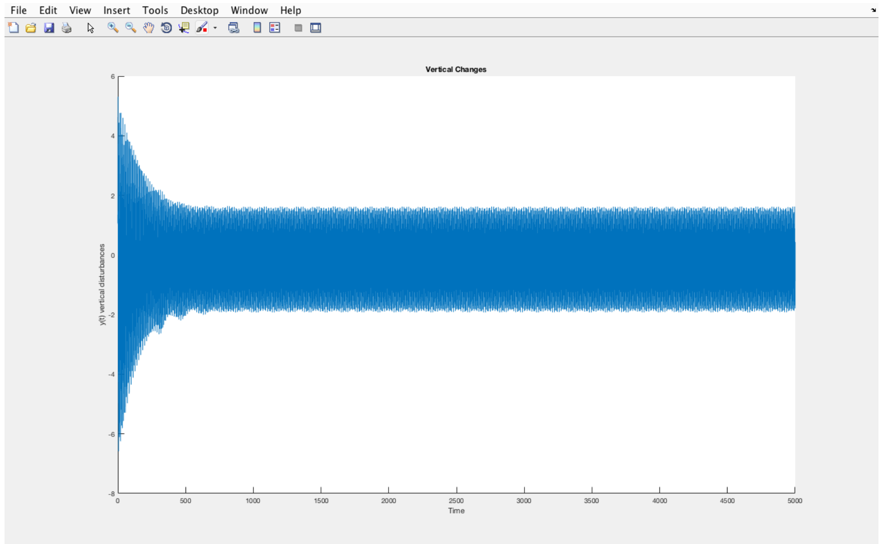

Figure 1: $y(t)$ vs t for the Tacoma Bridge Model when w = 80, initial $\theta = 0.001$

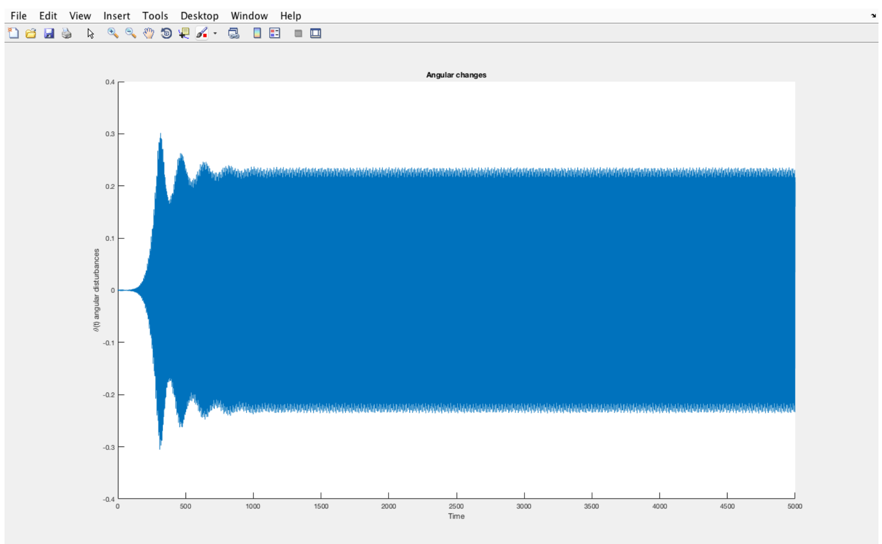

Figure 2: $\theta(t)$ vs t for the Tacoma Bridge Model when w = 80, initial $\theta = 0.001$

- Part III :
  In this case we want to find a wind speed, w, where the magnifying factor will be 100 or more. We can test for different values of w and from experiment find the value that corresponds to the desired magnifying factor.

  If we use the given initial values, meaning $[1, 0, 0.001, 0]$, we get an interesting case where we have a huge jump in the magnifying factor, moving from 87.142 to 205.794 *mph* very suddenly within the wind speed of $w = 59$ and $w = 60$ (as shown in figure 3)[Corresponding code: tacoma_part3.m].

```
Command Window
>> tacoma_part3
To run with default values, please enter 1, otherwise enter 2
1
 for w = 55
 Magnifying factor = 4.337486e+01
 for w = 56
 Magnifying factor = 5.072473e+01
 for w = 57
 Magnifying factor = 5.951247e+01
 for w = 58
 Magnifying factor = 7.082659e+01
 for w = 59
 Magnifying factor = 8.714222e+01
 for w = 60
 Magnifying factor = 2.057941e+02
 for w = 61
 Magnifying factor = 2.153998e+02
 for w = 62
 Magnifying factor = 2.225427e+02
 for w = 63
 Magnifying factor = 2.293843e+02
 for w = 64
 Magnifying factor = 2.333841e+02
 for w = 65
 Magnifying factor = 2.393425e+02
>>
```

Figure 3: The output for testing different values of wind speed, w, and the corresponding magnifying factors. Initial condition $= [1, 0, 0.001, 0]$

In the other case, we thought that it would be better to use the initial condition of $[0, 0, 0.001, 0]$, meaning $y(1,1)$ which is the initial placement of the bridge from the equilibrium, we place it at zero and thus we get a much better and more sensible result. Running the code (Corresponding code: tacoma_part3.m) we will get the following:

```
Command Window
>> tacoma_part3
To run with default values, please enter 1, otherwise enter 2
1
 for w = 55
 Magnifying factor = 4.307561e+01
 for w = 56
 Magnifying factor = 5.122299e+01
 for w = 57
 Magnifying factor = 6.155147e+01
 for w = 58
 Magnifying factor = 7.575393e+01
 for w = 59
 Magnifying factor = 1.004610e+02
 for w = 60
 Magnifying factor = 2.095814e+02
 for w = 61
 Magnifying factor = 2.185854e+02
 for w = 62
 Magnifying factor = 2.272185e+02
 for w = 63
 Magnifying factor = 2.327945e+02
 for w = 64
 Magnifying factor = 2.399577e+02
 for w = 65
 Magnifying factor = 2.455027e+02
>>
```

Figure 4: The output for testing different values of wind speed, w, and the corresponding magnifying factors. Initial condition $= [0, 0, 0.001, 0]$

As we can see in the outputs of the code, Figure 4, the first magnifying factor 100 or more corresponds with a wind speed of $w = 59$ and a magnifying factor of 100.4610. However, for the given initial values, we get the first magnifying factor over 100 to correspond to the wind speed $w = 60$. In both cases, for all the larger values of w, the magnifying factors keep on increasing. So it is safe to say that the first wind speed that will have a magnifying factor larger than 100 will be between 59 and 60. The values of w vs. Magnifying Factor were plotted in both cases, both have very similar shape and as mentioned before, the desired magnifications occur between 59 and 60 (Figures 5,6).

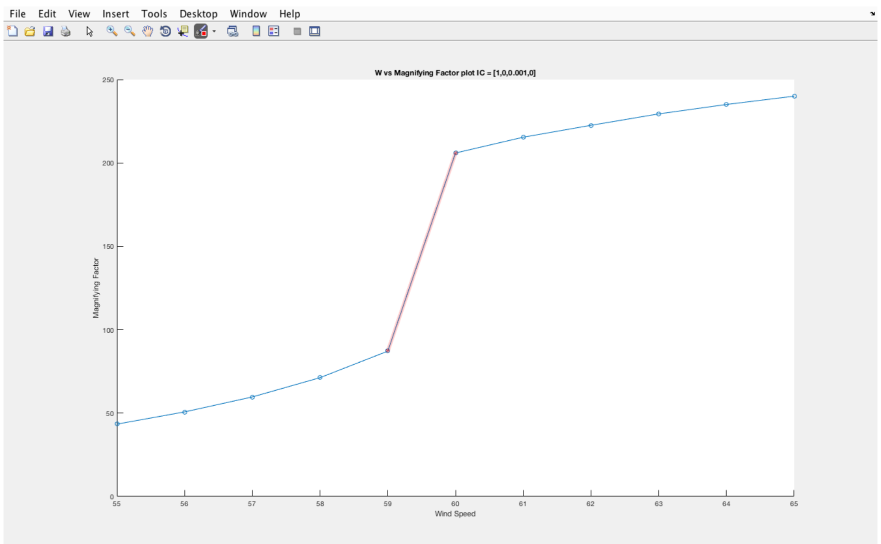

Figure 5: The Wind Speed versus Magnifying factors graph. Initial condition = $[1, 0, 0.001, 0]$

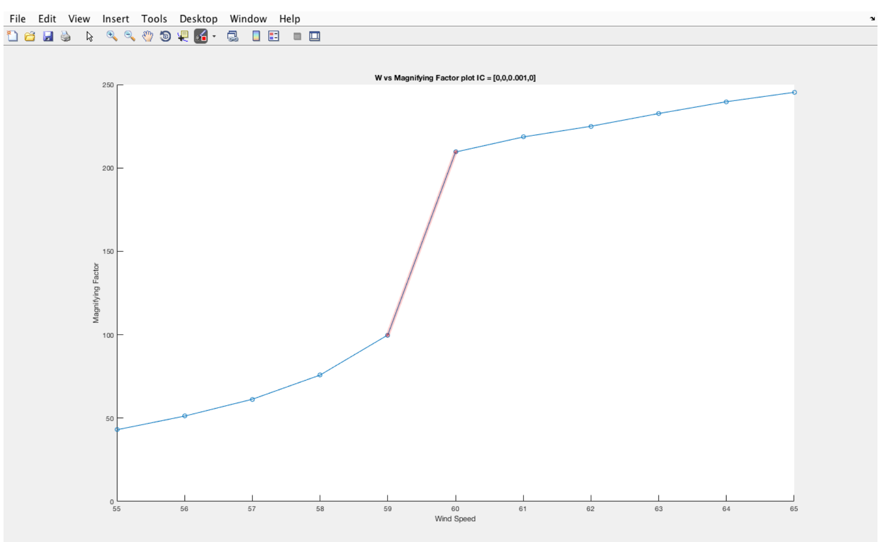

Figure 6: The Wind Speed versus Magnifying factors graph.Initial condition = $[0, 0, 0.001, 0]$

- Part IV:

In this part, we tried larger values of wind speed, w, to see if all extremely small angles lead to catastrophic size (corresponding code: tacoma_part4).

$w = 90, \theta = 0.000001$

We will first check this with $w = 90$ and $\theta = 0.000001$. In this case we can see that the behavior is stable, since the angular oscillations die out as time goes forward and the bridge oscillations also stabilize.

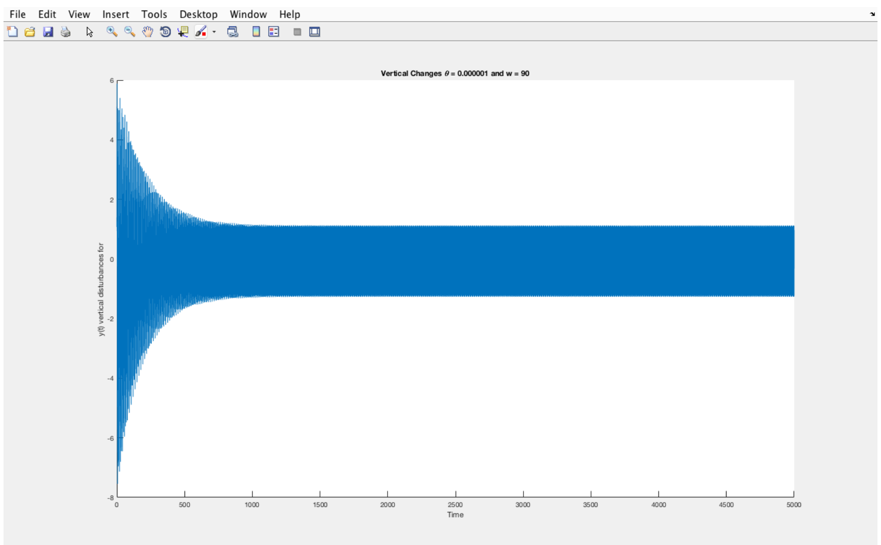

Figure 7: $y(t)$ vs t for the Tacoma Bridge Model when w = 90, initial $\theta = 0.000001$

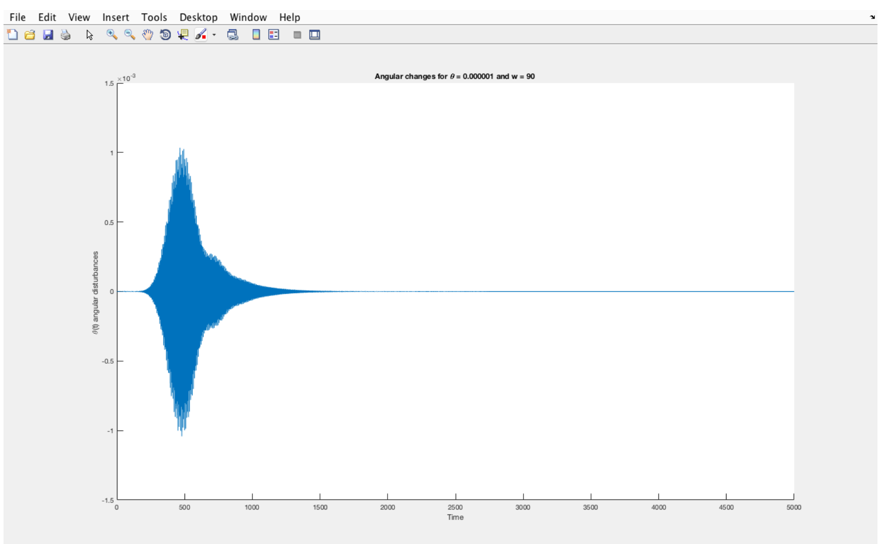

Figure 8: $\theta(t)$ vs t for the Tacoma Bridge Model when w = 90, initial $\theta = 0.000001$

$w = 100, \theta = 0.000001$

In this case we can see that the behavior is also stable, since the angular oscillations die out as time moves on, although it seems to last longer but eventually it dies out. Similarly, the bridge oscillations also stabilize.

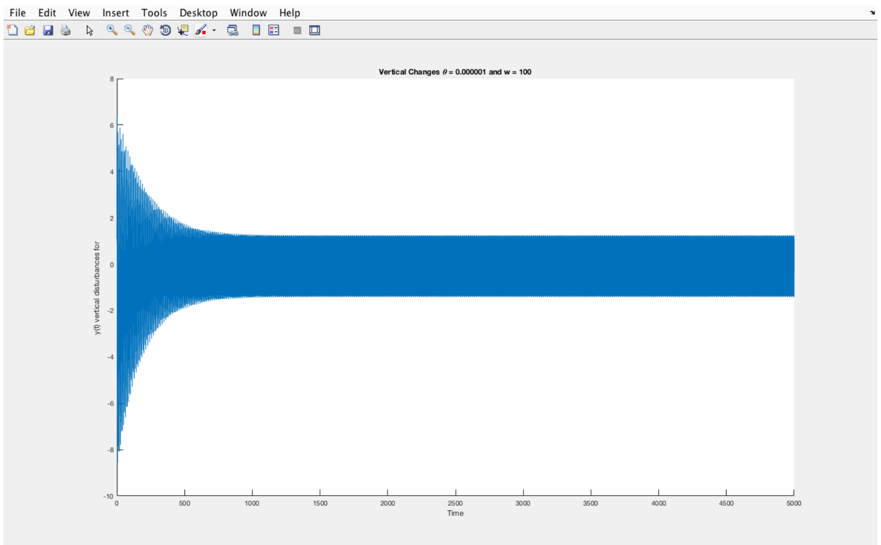

Figure 9: $y(t)$ vs t for the Tacoma Bridge Model when w = 90, initial $\theta = 0.000001$

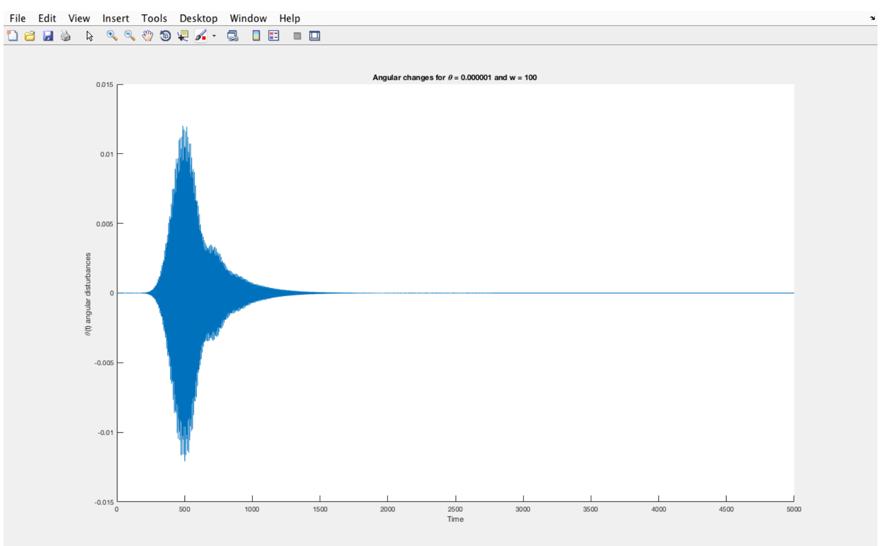

Figure 10: $\theta(t)$ vs t for the Tacoma Bridge Model when w = 100, initial $\theta = 0.000001$

$w = 100, \theta = 0.000001$

For these values we finally get the case that the bridge is unstable. The angular disturbances remain high in oscillation, very similar to the initial conditions in part 1. So we can see that even with extremely small θ we can get an unstable bridge, but for very large wind speeds.

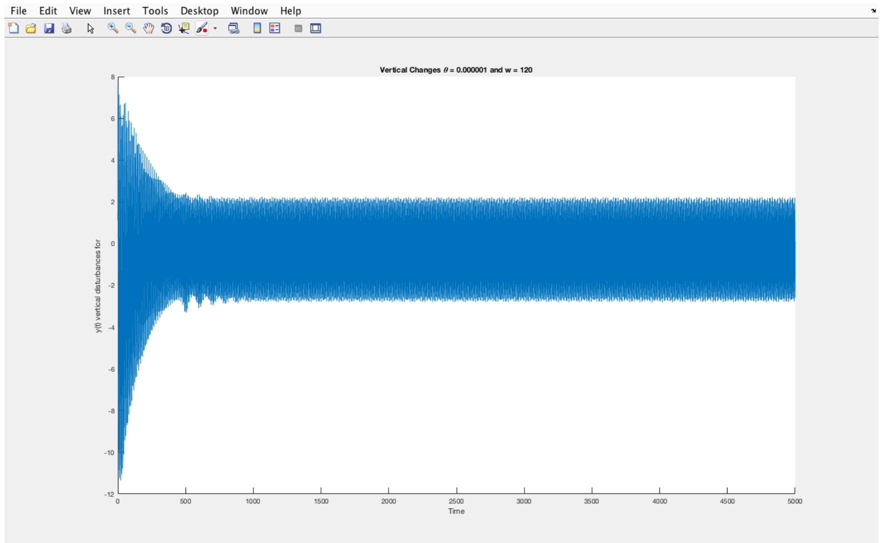

Figure 11: $y(t)$ vs t for the Tacoma Bridge Model when w $= 90$, initial $\theta = 0.000001$

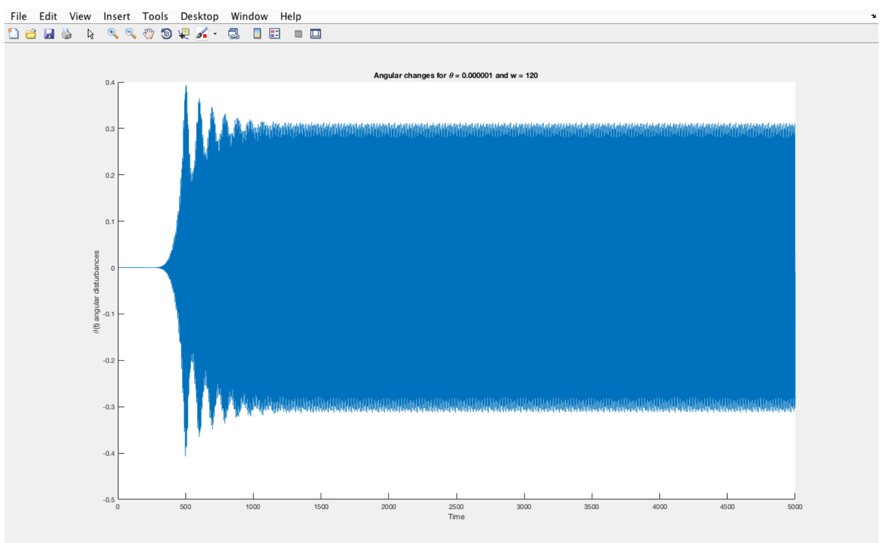

Figure 12: $\theta(t)$ vs t for the Tacoma Bridge Model when w $= 100$, initial $\theta = 0.000001$

$w = 90, \theta = 0.0001$

intuitively, this case should be similar to the original question where $\theta = 0.001$. In our simulation we see that for this case the angular oscillations do not die out and therefore the bridge becomes unstable. Since we have that for this value of $w = 90$ we have such instability, there is no need to study when $w = 100$ or larger, since the bridge will also be unstable for those values.

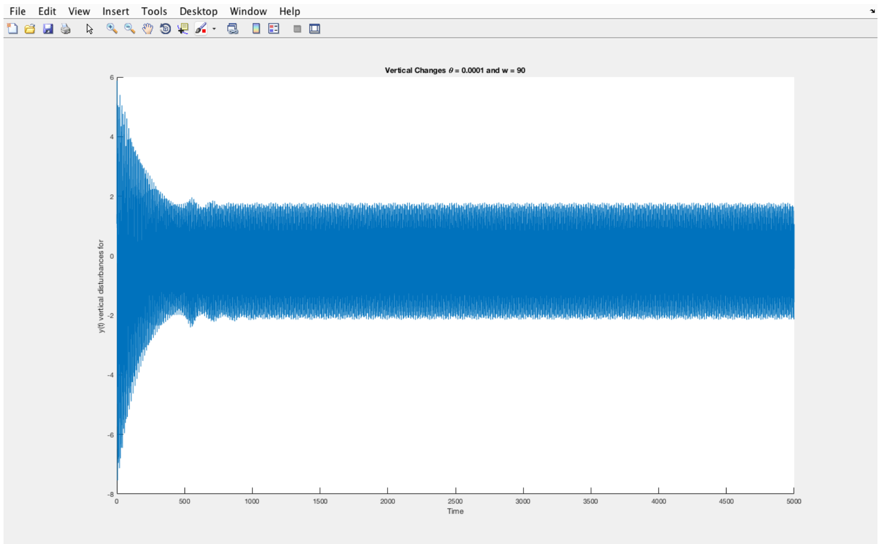

Figure 13: $y(t)$ vs t for the Tacoma Bridge Model when w $= 90$, initial $\theta = 0.0001$

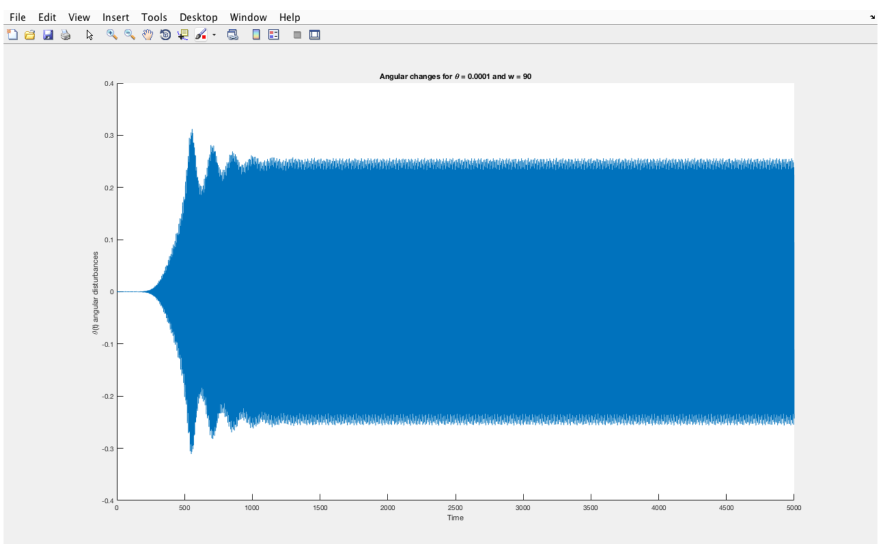

Figure 14: $\theta(t)$ vs t for the Tacoma Bridge Model when w $= 90$, initial $\theta = 0.0001$

As we can note, not for all extremely values of θ we can see chaotic behavior. In the first part that we used a very tiny $\theta = 0.000001$, we saw that the bridge was stable and the angular oscillations did die out before the wind is very fast, such as when $w = 120$ where the angular disturbances remained in high oscillation and the bridge was unstable. On the other hand, as we increase the θ, for example the second testing case where $\theta = 0.0001$ we saw instability for large wind speeds such as $w = 90$.

- Part V:
  In this part, we will double the damping coefficent from $d = 0.01$ to $d = 0.02$ and see the effect of this change to the critical wind speed. So we first run our code again to see what changes this has made to the problem with the given initial conditions and the wind speed of $w = 80$ (Corresponding code: tacoma_part5.m).

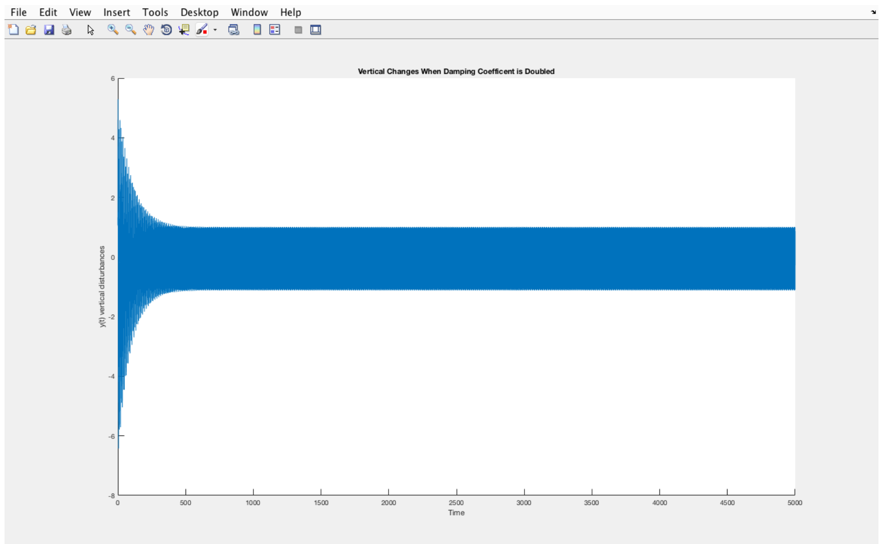

Figure 15: $y(t)$ vs t for the Tacoma Bridge Model when damping coefficient is doubled, $d = 0.02$

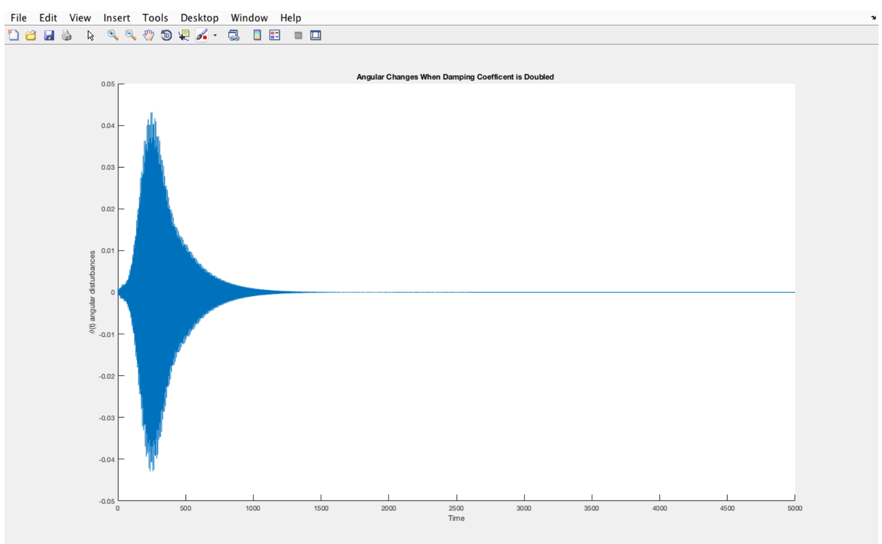

Figure 16: $\theta(t)$ vs t for the Tacoma Bridge Model when damping coefficient is doubled, $d = 0.02$

As we can see, if the damping coefficient is doubled, the bridge would eventually stabilize and the angular disturbances would die out.

Now to check for the critical w, we rerun the code from before, now saved as (Corresponding code: tacoma_part6.m), but this time from $w = 50 : 100$ and see what values we achieve this time.

As seen in the figures 14, 15, increasing the damping coefficient will make the angular oscillations die out and thus the bride being stable. The below graph shows the magnifying factor vs. the wind speed after doubling the damping coefficient.

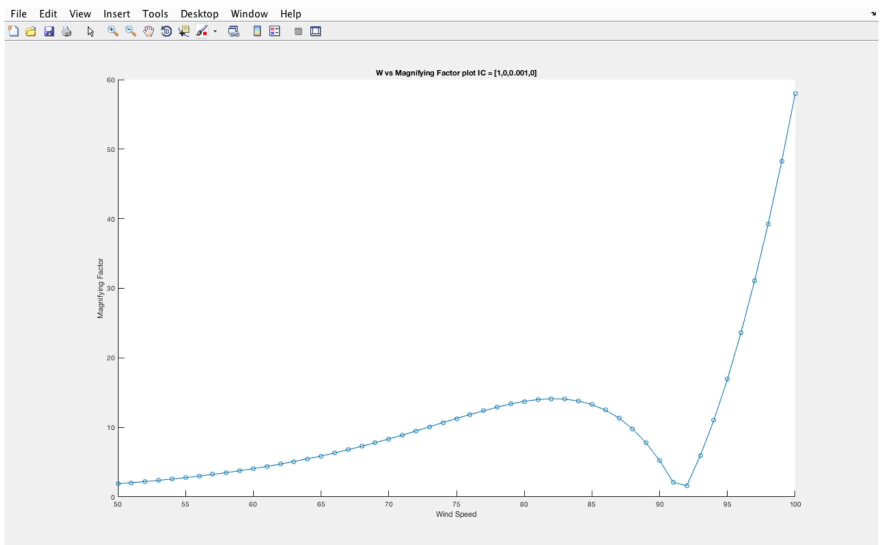

Figure 17: The Wind Speed versus Magnifying factors when damping coefficient , d, is doubled to 0.02

In figure 17 above, we could see that doubling the damping coefficient will keep the magnifying below 60 even when the wind speed is $w = 100$, and below 20 for up to $w = 95$. Again, doubling the damping coefficient will increase the stability of the bridge in a considerable amount.

A few design suggestion could be :

(1) The engineers should have Built the bridge in a way that the wind can be distributed around the bridge instead of acting on it directly as a force

(2) It could have been stiffer by using thicker cables with better suspension (more like a spring rather than a string).

(3) They could have used materials that have higher damping which could have helped the bridge to be more stable and less susceptible to torsion

(4) The truss under the bridge should have been deeper and stringer to keep the bridge more stable

Theoretical Question 1

In this problem we want to show that how we could apply the given relation

$$y(x+h) = 2y(x) - y(x-h) + \frac{h^2}{12}\left[y''(x+h) + 10y''(x) + y''(x-h)\right] + \mathcal{O}(h^6)$$

to a finite difference method in order to obtain an approximate numerical solution of the following boundary-value problem $y'' = f(x)y + g(x) \qquad x \in [a,b]\ y(a) = \alpha$ and $y(b) = \beta$

First, we want to do the proper substitutions in terms of numeric approximation. So let

$$w_i = y(x) \qquad w_{i-1} = y(x-h) \qquad w_{i+1} = y(x+h)$$

and using the boundary condition we will have that

$$w_0 = \alpha \qquad w_{N+1} = \beta \qquad \text{for } N \text{ the last iteration}$$

Also, we know that $y'' = f(x)y + g(x) = f(x)w_i + g(x)$

. Now replacing these equivalences in the original equation:

$$w_{i+1} = 2w_i - w_{i-1} + \frac{h^2}{12}\left[y''(x+h) + 10y''(x) + y''(x-h)\right] + \mathcal{O}(h^6)$$

$$0 = -w_{i+1} + 2w_i - w_{i-1} + \frac{h^2}{12}[f(x+h)w_{i+1} + g(x+h) + 10(f(x)w_i + g(x)) + f(x-h)w_{i-1} + g(x-h)]$$

$$0 = -w_{i+1} + 2w_i - w_{i-1} + [\frac{h^2}{12}f(x+h)w_{i+1}] + \frac{h^2}{12}g(x+h) + [\frac{10h^2}{12}f(x)w_i] + \frac{10h^2}{12}g(x) + [\frac{h^2}{12}f(x-h)w_{i-1}] + \frac{h^2}{12}g(x-h)$$

Combining like terms we obtain :

$$-[\frac{h^2}{12}(g(x+h) + 10g(x) + g(x-h))] = w_{i+1}[-1 + \frac{h^2}{12}f(x+h)] + w_i[2 + \frac{5h^2}{6}f(x)] + w_{i-1}[-1 + \frac{h^2}{12}f(x-h)]$$

Now to solve this, we can turn this into a matrix equation of the form

$$[A][W] = [B]$$

So we would have (not yet turning $x+h, x-h$ in terms of x_i):

$$A = \begin{bmatrix} 2 + \frac{5h^2}{6}f(x) & -1 + \frac{h^2}{12}f(x-h) & 0 & 0 & \dots & 0 \\ 0 & -1 + \frac{h^2}{12}f(x+h) & 2 + \frac{5h^2}{6}f(x) & -1 + \frac{h^2}{12}f(x-h) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & 0 & -1 + \frac{h^2}{12}f(x+h) & 2 + \frac{5h^2}{6}f(x) & -1 + \frac{h^2}{12}f(x-h) \\ 0 & 0 & \dots & 0 & -1 + \frac{h^2}{12}f(x+h) & 2 + \frac{5h^2}{6}f(x) \end{bmatrix}$$

Where A is an $N \times N$ tridiagonal matrix. (N is the number of steps we desire).

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}$$

and

$$B = \begin{bmatrix} -[\frac{h^2}{12}(g(x+h) + 10g(x) + g(x-h))] \\ -[\frac{h^2}{12}(g(x+h) + 10g(x) + g(x-h))] \\ \vdots \\ -[\frac{h^2}{12}(g(x+h) + 10g(x) + g(x-h))] \end{bmatrix}$$

But since we are on a grid for solving the finite differences, we must also have x's in grids as well. As mentioned in our textbook in pages 700 and 701, if we expand $y(x_{i+1})$ about x_i, we will find that $y(x_{i+1}) = y(x+h)$, $y(x_i) = y(x_i)$ and $y(x_{i-1}) = y(x-h)$.

Now for the ease of notation, let

$$p(x_i) = f(x_{i+1}), q(x_i) = f(x_{i-1}, r(x) = f(x_i)$$

and let

$$G(x_i) = -[\frac{h^2}{12}(g(x_{i+1}) + 10g(x_i) + g(x_{i-1})]$$

Now knowing these equivalences, we will have:

$$A = \begin{bmatrix} 2 + \frac{5h^2}{6}r(x_1) & -1 + \frac{h^2}{12}q(x_1) & 0 & 0 & \dots & 0 \\ 0 & -1 + \frac{h^2}{12}p(x_2) & 2 + \frac{5h^2}{6}r(x_i) & -1 + \frac{h^2}{12}q(x_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & 0 & -1 + \frac{h^2}{12}p(x_{N-1}) & 2 + \frac{5h^2}{6}r(x_{N-1}) & -1 + \frac{h^2}{12}q(x_{N-1}) \\ 0 & 0 & \dots & 0 & -1 + \frac{h^2}{12}P(x_N) & 2 + \frac{5h^2}{6}r(x_N) \end{bmatrix}$$

and

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}$$

$$B = \begin{bmatrix} G(x_1) - (-1 + \frac{h^2}{12} q(x_1)) w_0 \\ G(x_2) \\ G(x_3) \\ \vdots \\ G(x_N) - (2 + \frac{5h^2}{6} r(x_N)) w_{N+1} \end{bmatrix}$$

and now we must solve $A \cdot W = B$.

Theoretical Question 2

In this problem, we are trying to analyze the stability of the multi-step method:

$$w_{n+1} = 4w_n - 3w_{n-1} - 2hf(t_{n-1}, w_{n-1}), n > 1$$

So we would like to find the characteristic polynomial of this method, and for that ,we would like to do is to put this in the general format:

$$w_{i+1} = a_{m-1} w_i + a_{m-2} w_{i-1} + \cdots + a_0 w_{i+1-m}$$

so we can use the root condition and analyze the stability. We must find all the roots for :

$$P(\lambda) = \lambda^m - a_{m-1}\lambda^{m-1} - \cdots - a_1 \lambda - a_0 = 0$$

Now since we have that $w_{n+1} = 4w_n - 3w_{n-1} - 2hf(t_{n-1}, w_{n-1}), n > 1$, we can easily find that $m = 2$ with $a_{m-1} = a_1 = 4$ and $a_{m-2} = a_0 = -3$. So our characteristic polynomial will be:

$$0 = \lambda^2 - 4\lambda + 3 = (\lambda - 1)(\lambda - 3) = 0$$

$\implies \lambda = 3, 1.$

Now we know by the root condition that if $|\lambda_i| \leq 1$ for all $i = 1, \cdots N-2$ then the method is stable. But since we do not have both roots being less than or equal to 1, then we know that our method is unstable. The region R of absolute stability for a one-step method is $R = \{h\lambda \in C : |Q(h\lambda)| < 1\}$, where $Q(z, h\lambda) = (1 - h\lambda b_m) z_m - (a_{m-1} + h\lambda b_{m-1}) z_{m-1} - \cdots - (a_0 + h\lambda b_0)$. Now following this analysis, we can get the following graph for the region of absolute stability:

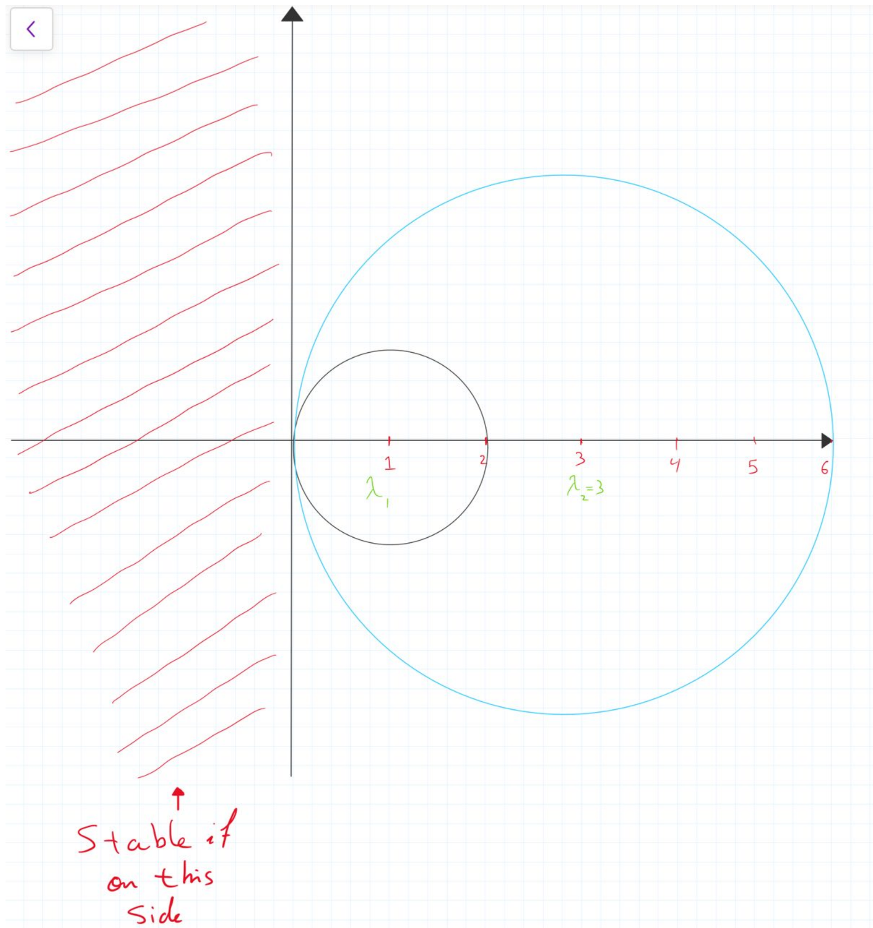

Figure 18: The solution to the burgers equation with $h = k = 0.1$ for the function $f = sin(2\pi x)$ and a viscosity constant of D = 0.2

Once again, we can see that the method is unstable, since none of the λ-balls fall within the left of the plane (marked in red in figure 18).

Theoretical Question 3 We know that the shooting methods tries to solve a Boundary Value problem (BVP) by an Initial Value Problem (IVP) that has the same solution. We first guess the slope from the first point, and then the IVP that is with this guess will be solved and compared to the later boundary of the BVP. By trial and error we keep on improving the slopes until we eventually get both to math and thus have solved the BVP.

For this problem, we have that

$$y'' = xy' + y + 2cos(x) \qquad y(0) = 1, y(1) = 9 \qquad (1)$$

$$y'' = y^2 - x + yx \qquad y(0) = 1, y(1) = 3 \qquad (2)$$

For (1), we can see that since the differential equation is in the form $y'' = p(x)y' + q(x)y + r(x)$ with $y(a) = \alpha andy(b) = \beta$, then the Differential equation is linear and since : (i) $x, 1$, and $cos(x)$

are continuous on $[0, 1]$, (ii) $q(x) > 0$ on $[0, 1]$ we know that we will have a unique solution.

Now we have to use some method such as Euler's, Runge-Kutta etc. to solve this but the methods will require initial value problems. So we can reconstruct the problem by introducing $z = \frac{dy}{dx}$, with $z(0) = C$ (this is our initial guess). So now

$$\frac{dz}{dx} = xz + y + 2cos(x) \qquad y(0) = 1$$

$$\frac{dy}{dx} = z \qquad y'(0) = z(0) = C$$

And from here, we could choose a method that uses initial value problems to solve the boundary value problem. To check if our guess was correct or not, we need to get that after our approximation, say w is the approximations, then $w(1) = 9$. If not, then we need to change our guess for $y'(0)$ to a different value.

Now that we have reduced our problem into first order IVP's, we could use our favorite method, such as Runge-Kutta, and continue the approximation.

For (2) equation, we have that

$$y'' = y^2 - x + yx \qquad y(0) = 1, y(1) = 3 \qquad (2)$$

. in this case, this is a non-linear boundary value problem. For non-linear boundary value problems, we need to follow the same steps as the linear shooting except that we need to use Newton's or Secant method to find the next guess that we need in our approximation. But for the initial value condition, we can do the following:

We again want to reduce the order of the problem to a first order, so we can first introduce a new variable $z = \frac{\partial y'}{\partial t}(x, t)$, but before that we want to see what we would get.
Now we can pick either Newton's method or secant method to solve this problem; and for that $(dy/dt)(b, t)$ when $t = t_{k1}$, so we first take the partial derivative of the rewritten IVP

$$y''(x, t) = f(x, y(x, t), y'(x, t)), \qquad a \leq x \leq b, \qquad y(a, t) = \alpha \qquad y'(a, t) = t$$

with respect to t. In our case it would be :

$$y'' = y^2 - x + yx \qquad y(0) = 1, y(1) = 3 \qquad (2)$$

Now this implies that:

$$\frac{\partial y''}{\partial t}(x, t) = \frac{\partial f}{\partial t}(x, y(x, t), y'(x, t))$$
$$= \frac{\partial f}{\partial x}(x, y(x, t), y'(x, t))\frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}(x, y(x, t), y'(x, t))\frac{\partial y}{\partial t}(x, t)$$

$$+\frac{\partial f}{\partial y'}(x, y(x,t), y'(x,t))\frac{\partial y'}{\partial t}(x,t)$$

Due to the independence of the variables x, t we get $\partial x/\partial t = 0$, and the equation simplifies to

$$\frac{\partial y''}{\partial t}(x,t) = \frac{\partial f}{\partial t}(x, y(x,t), y'(x,t))\frac{\partial y}{\partial t}(x,t) + \frac{\partial f}{\partial y'}(x, y(x,t), y'(x,t))\frac{\partial y'}{\partial t}(x,t)$$

$$z''(x,t) = \frac{\partial f}{\partial y}(x, y, y')z(x,t) + \frac{\partial f}{\partial y'}(x, y, y')z'(x,t)$$

and now our initial guess would be for $\frac{\partial y'}{\partial t}(0,t) = C$

Now reducing the order in our problem with z we get:

$$z''(x,t) = (2y + x)z(x,t) + 0$$

with the intial condition of $y(0) = 1$ and $z(0) = C$

Now if want to apply the Newton's method, we would have

$$t_k = t_{k-1} - \frac{y(b, t_{k-1}) - \beta}{\frac{dy}{dt}(b, t_{k-1})} = t_{k-1} - \frac{y(b, t_{k-1}) - \beta}{z(b, t_{k-1})}$$

and at the last step we should check it $z(b) = z(1) = 3$, if not we need to re-asses our initial guess and do the same process again.

**** To run all the provided code, please press "run" in MATLAB and follow the prompt ****

Computer Project 2

In this problem, we are trying to experiment with the differential equations and values to explore different phenomenas from this model.

- part I :

  (a) We are trying to find the solutions for the given Burgers equations with $k = h = 0.1$. To use the given code, we must first find the values for M, N which are :

$$M = \frac{x_r - x_l}{h} = \frac{1 - 0}{0.1} = 10$$

$$N = \frac{t_e - t_b}{k} = \frac{1 - 0}{0.1} = 10$$

  Now running code (corresponding code: burgers_a.m) with the given values we obtain the following solution :

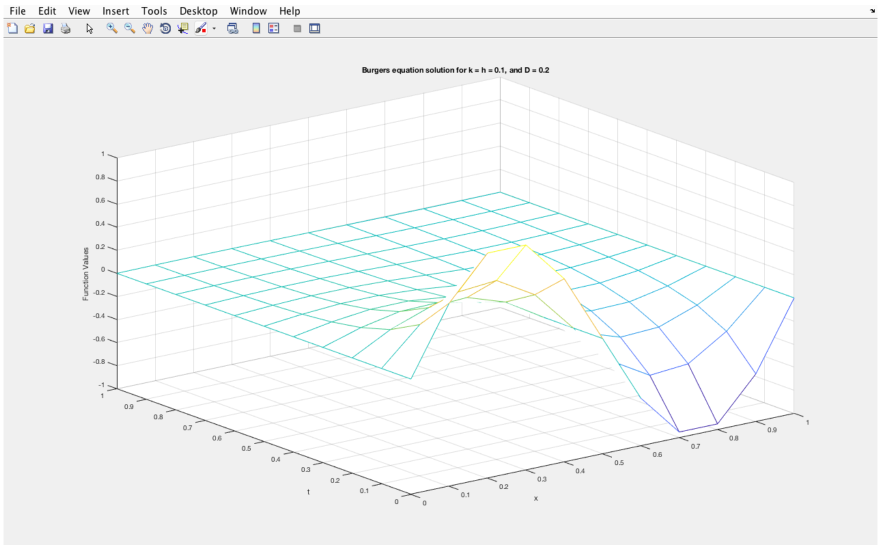

Figure 19: The solution to the burgers equation with $h = k = 0.1$ for the function $f = sin(2\pi x)$ and a viscosity constant of D = 0.2

(b) Now in this part, we want to find the approximation solution for the step sizes $h = k = 0.02$, which intuitively should give us a more accurate result on more grids. To find M and N we do the same calculations, this time with h and k being 0.02:

$$M = \frac{x_r - x_l}{h} = \frac{1-0}{0.02} = 50$$
$$N = \frac{t_e - t_b}{k} = \frac{1-0}{0.02} = 50$$

and as predicted intuitively, by running the code (corresponding code: burgers_b.m), we get a much better approximation with on a grid much more divided and the shape being better visible (Figure 20)

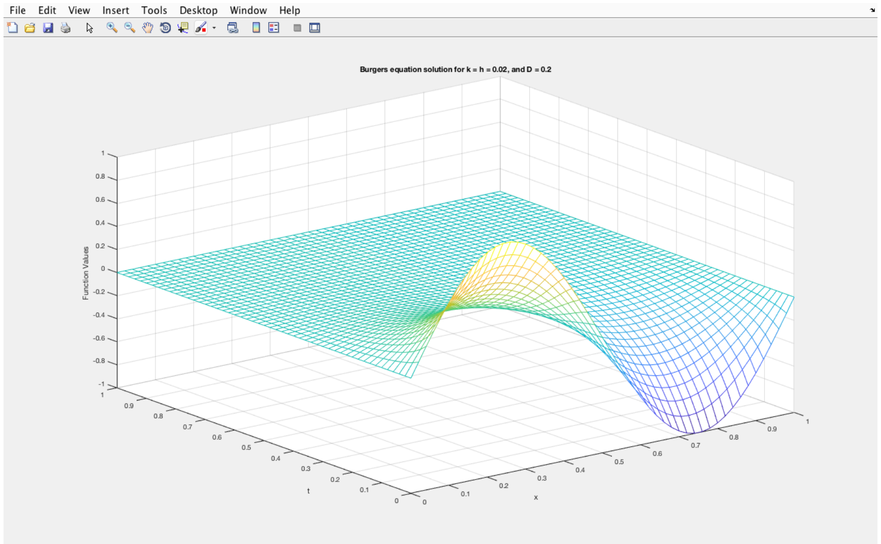

Figure 20: The solution to the burgers equation with $h = k = 0.1$ for the function $f = sin(2\pi x)$ and a viscosity constant of D $= 0.2$

As we can note, the solutions tend to zero as the time increases, so the equilibrium would be 0 as t increases (as evident by both graphs on the t axis).

- Part II: (a) For this problem, we first want to show that

$$u(x,t) = \frac{2D\beta\pi e^{-D\pi^2 t}\sin\pi x}{\alpha + \beta e^{-D\pi^2 t}\cos\pi x}$$

is a solution of

$$\begin{cases} u_t + uu_x = Du_{xx} & \\ u(x,0) = \frac{2D\beta\pi\sin\pi x}{\alpha+\beta\cos\pi x} & for 0 \leq x \leq 1 \\ u(0,t) = 0 & for all t \geq 0 \\ u(1,t) = 0 & for all t \geq 0 \end{cases}$$

In order to show this, we first want to look at the value of the solution at each specific boundary, meaning :

$$u(0,t) = \frac{2D\beta\pi e^{-D\pi^2 t}\sin\pi(0)}{\alpha+\beta e^{-D\pi^2 t}\cos\pi(0)} = \frac{0}{\alpha+\beta e^{-D\pi^2 t}} = 0$$

$$u(x,0) = u(x,t) = \frac{2D\beta\pi e^{-D\pi^2(0)}\sin\pi x}{\alpha+\beta e^{-D\pi^2(0)}\cos\pi x} = \frac{2D\beta\pi\sin\pi x}{\alpha+\beta\cos\pi x}$$

$$u(1,t) = \frac{2D\beta\pi e^{-D\pi^2 t}\sin\pi(1)}{\alpha+\beta e^{-D\pi^2 t}\cos\pi(1)} = \frac{0}{\alpha+\beta e^{-D\pi^2 t}} = 0$$

Our next step is to find u_x, $u_x x$ and u_t :

To find u_x, we will differentiate $u(x,t)$ with respect to u_x :
$\frac{\partial}{\partial x}u(x,t) = \frac{(\alpha+\beta e^{-D\pi^2 t}\cos\pi x)\pi(2D\beta\pi e^{-D\pi^2 t}\cos\pi x)-(\beta e^{-D\pi^2 t}-\sin\pi x)(2D\beta\pi e^{-D\pi^2 t}\sin\pi x)(\pi)}{\alpha+\beta De^{-D\pi^2 t}\cos\pi x}$
after simplifying we get :

$u_x = \frac{(2\pi^2 D\alpha\beta e^{-D\pi^2 t}\cos\pi x)+(2\pi^2 D\beta^2 e^{-2D\pi^2 t}sin^2\pi x)+(2\pi^2 D\beta^2 e^{-2D\pi^2 t}\cos^2\pi x)}{(\alpha+\beta e^{-D\pi^2 t\cos\pi x})^2}$

combining alike terms :

$u_x = \frac{(2\pi^2 D\alpha\beta e^{-D\pi^2 t}cos\pi x)+(2\pi^2 D\beta^2 e^{-2D\pi^2 t})}{(\alpha+\beta e^{-D\pi^2 t}cos\pi x)^2}$
and finally we obtain :

$$u_x = \frac{(2\pi^2 D\beta e^{-2D\pi^2 t})(\beta+\alpha e^{D\pi^2 t}\cos\pi x)}{(\alpha+\beta e^{-D\pi^2 t}cos\pi x)^2}$$

Now to find u_{xx} we take the partial derivative of u_x:
$u_{xx} = \frac{\partial}{\partial x}[\frac{(2\pi^2 D\beta e^{-2D\pi^2 t})(\beta+\alpha e^{D\pi^2 t}\cos\pi x)}{(\alpha+\beta e^{-D\pi^2 t}cos\pi x)^2}]$
$u_{xx} = (2\pi^2 D\beta e^{-2D\pi^2 t})[\frac{(\alpha+\beta e^{-D\pi^2 t}\cos\pi x)^2(-\pi\alpha e^{D\pi^2 t}\sin\pi x)-2(\beta+\alpha e^{D\pi^2 t}\cos\pi x)(\alpha+\beta e^{-D\pi^2 t}\cos\pi x)(-\pi\beta e^{-D\pi^2 t}\sin\pi x)}{(\alpha+\beta e^{-D\pi^2 t}\cos\pi x)^4}]$
and after lots of simplifying we obtain :

$$u_{xx} = (2\pi^2 D\beta e^{-2D\pi^2 t})\{\frac{(-\pi e^{D\pi^2 t}\sin\pi x)((\alpha^2+\alpha\beta e^{D\pi^2 t}\cos\pi x)-2(\beta^2+\alpha\beta e^{D\pi^2 t}\cos\pi x))}{(\alpha+\beta e^{D\pi^2 t}\cos\pi x)^3}\}$$

Finally, we want to find u_t :

To obtain u_t, differentiate u partially with respect to t. Thus,
$u_t = \frac{\partial}{\partial t}\{\frac{-D\beta\pi e^{-D\pi^2 t}\sin\pi x}{\alpha+\beta e^{-D\pi^2 t}\cos\pi x}\}$

$= \frac{\left(\alpha+\beta e^{-D\pi^2 t}\cos\pi x\right)\left(2D\beta\pi e^{-D\pi^2 t}\sin\pi x\right)\left(-D\pi^2\right)-(2D\beta\pi e^{-D\pi^2 t}\sin\pi x)(\beta e^{-D\pi^2 t}\cos\pi x)(-D\pi^2)}{(\alpha+\beta e^{-D\pi^2 t}\cos\pi x)^2}$

$= \frac{-\alpha(2D^2\pi^2\beta\pi e^{-D\pi^2 t}\sin\pi x)-(\beta e^{-D\pi^2 t}\cos\pi x)(2D^2\pi^2\beta\pi e^{-D\pi^2 t}\sin\pi x)}{(\alpha+\beta e^{-D\pi^2 t}\cos\pi x)^2}$

$+\frac{(2D^2\pi^2\beta\pi e^{-D\pi^2 t}\sin\pi x)(\beta e^{-D\pi^2 t}\cos\pi x)}{(\alpha+\beta e^{-D\pi^2 t}\cos\pi x)^2}$
and finally we get :

$$= \frac{-\alpha(2D^2\pi^2\beta\pi e^{-D\pi^2 t}\sin\pi x)}{(\alpha+\beta e^{-D\pi^2 t}\cos\pi x)^2}$$

Add u_t and uu_x. This gives:
$u_t + uu_x = \frac{-\alpha(2D^2\pi^2\beta\pi e^{-D\pi^2 t}\sin\pi x)}{(\alpha+\beta e^{-D\pi^2 t}\cos\pi x)^2} + \left(\frac{2D\beta\pi e^{-D\pi^2 t}\sin\pi x}{\alpha+\beta e^{-D\pi^2 t}\cos\pi x}\right)\frac{(2\pi^2 D\beta e^{-D\pi^2 t})(\beta+\alpha e^{-D\pi^2 t}\cos\pi x)}{(\alpha+\beta e^{-D\pi^2 t}\cos\pi x)^2}$

$$= (2\pi^2 D^2 \beta e^{-D\pi^2 t}) \left(\frac{(-\pi e^{-D\pi^2 t} \sin \pi x)(\alpha(\alpha+\beta e^{-D\pi^2 t} \cos \pi x) - 2\beta(\beta+\alpha e^{D\pi^2 t} \cos \pi x))}{(\alpha+\beta e^{-D\pi^2 t} \cos \pi x)^3} \right) \text{ and we get :}$$

$$= D(2\pi^2 D\beta e^{-D\pi^2 t}) \left(\frac{(-\pi e^{-D\pi^2 t} \sin \pi x)(\alpha(\alpha + \beta e^{-D\pi^2 t} \cos \pi x) - 2\beta(\beta + \alpha e^{D\pi^2 t} \cos \pi x))}{(\alpha + \beta e^{-D\pi^2 t} \cos \pi x)^3} \right)$$

$$= Du_{xx}$$

so since we got that $u_t + uu_x = Du_{xx}$, we can confirm that the $u(x,t)$ is the solution to the Burger equation.

(b) in this part, we want to approximate a solution to the Burgers equations and then plot the errors. First we will try to run the code (corresponding code: burgers_part2.m) with $k = \frac{1}{16}$, $h = 0.01$ and $D = 0.2$. So we can find M, N by doing the following :

$$M = \frac{x_r - x_l}{h} = \frac{1-0}{0.01} = 100$$
$$N = \frac{t_e - t_b}{k} = \frac{1-0}{\frac{1}{16}} = 16$$

Running the code will give us the following approximation:

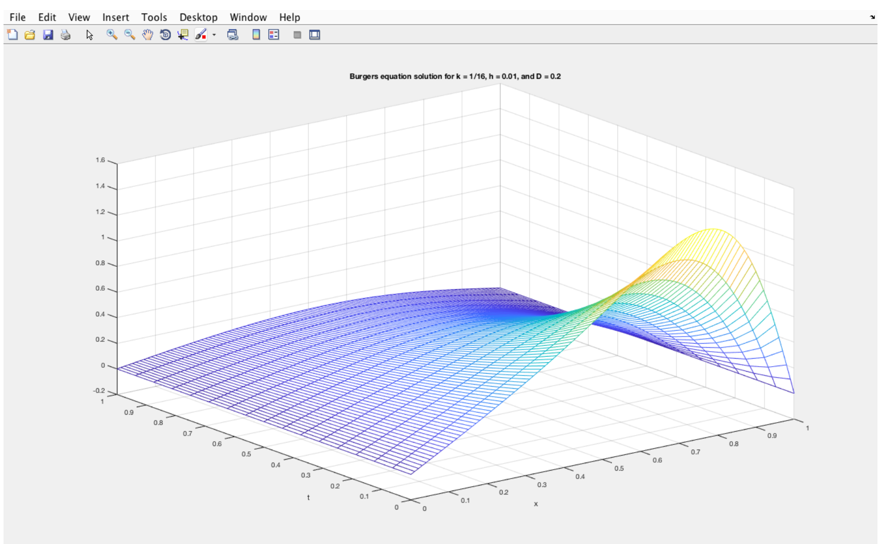

Figure 21: The solution for the Burger's equation with $k = 1/16$ and $h = 0.01$ and $D = 0.2$

Now we want to plot the error vs. the actual value at $x = \frac{1}{2}$ and $t = 1$ as the values of k decrease, where $k = 2^{-p}$ and $p = 4, 5, 6, 7, 8$. intuitively, the more we increase the number of time steps (decrease k), the less our error should be. Indeed, this is confirmed by our Figure 21.

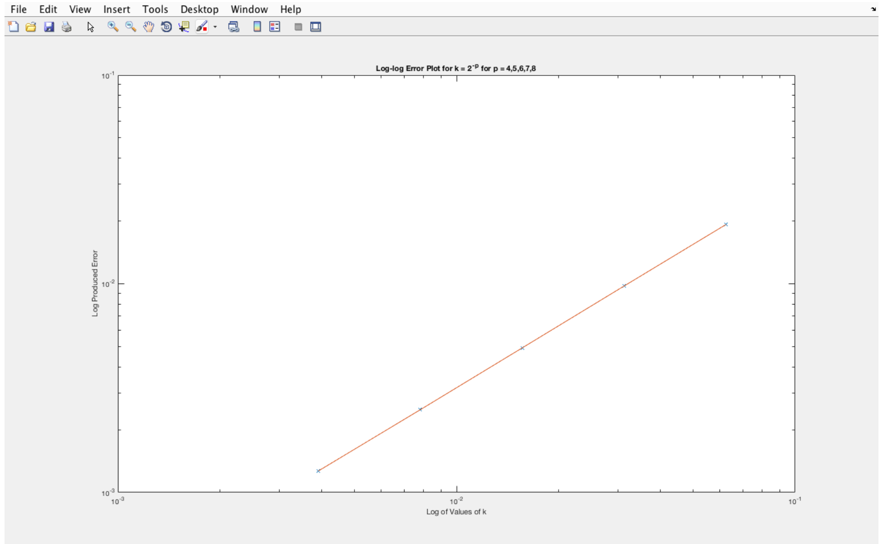

Figure 22: The log-log plot of the error as a function of k from the approximation and the exact solution at $x = 1/2$ and $t = 1$

As we can see in the figure above, as we iterate through $k = \{\frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}, \frac{1}{256}\}$, the error (as a function of k) decreases in a first order manner, specially that we kept h (the x space steps) the same throughout the code.