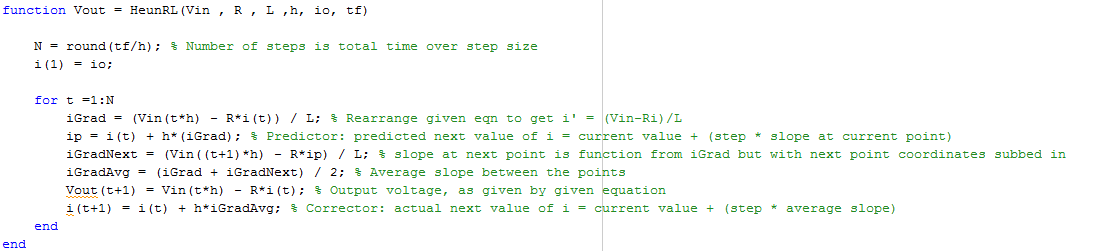
**EE2-08C: Numerical Analysis of ODEs/PDEs using Matlab**

1. **RL Circuit**

The following is the script HeunRL.m, which implements the Heun method for a function, Vin, passed to it:



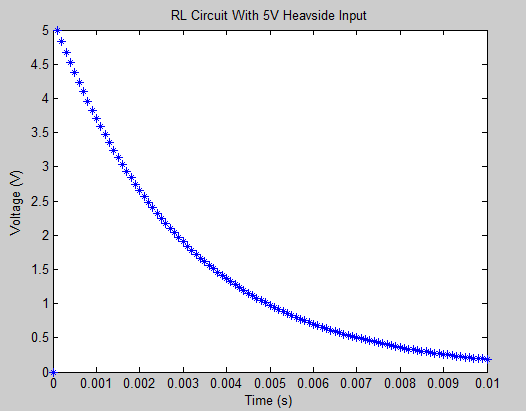
The function implements the improved Euler method as so:

1. Find the number of steps we will need to take by dividing the total length of x-axis we will be working on (i.e. time) by the step size
2. Rearrange the ODE given to us to get an equation for the gradient at the current point
3. Get a predicted next y-axis value corresponding to our next time point using the Euler method
4. Use the equation from **ii.** to get a prediction of the gradient at the next point using the next time value and our predicted y-axis value from **iii.**
5. A better estimate for the gradient between the current and next values is the average of their gradients
6. The voltage across the inductor is the input voltage minus the voltage across the resistor
7. The next y-axis value is the current value plus the step size multiplied by the average gradient from **v.**

I tested this function with various different input signals. These are the results:

**Input: Heaviside – h = 0.001, tf = 0.01**

With a 5V Heaviside input to the RL circuit, the output voltage varies as shown in the graph:



Our differential equation is:

http://latex.codecogs.com/gif.latex?L%5Cfrac%7Bd%7D%7Bdt%7D%20i%28t%29%20&plus;%20Ri%28t%29%20%3D%20V_%7Bin%7D%28t%29

Taking the Laplace transform gives:

http://latex.codecogs.com/gif.latex?LsI%28s%29&plus;LI%280%29%20&plus;%20RI%28s%29%20%3D%20V_%7Bin%7D%28s%29

Since i(0) = 0, this simplifies to:

http://latex.codecogs.com/gif.latex?LsI%28s%29&plus;%20RI%28s%29%20%3D%20V_%7Bin%7D%28s%29

http://latex.codecogs.com/gif.latex?I%28s%29%28Ls&plus;%20R%29%20%3D%20V_%7Bin%7D%28s%29

http://latex.codecogs.com/gif.latex?I%28s%29%20%3D%20%5Cfrac%20%7BV_%7Bin%7D%28s%29%7D%7B%28Ls&plus;%20R%29%7D

Also, we know that the voltage across an inductor is the product of the current and the reactance, and the transform of the reactance is *sL*:

http://latex.codecogs.com/gif.latex?V_%7BL%7D%28s%29%20%3D%20I%28s%29X%28s%29%20%3D%20I%28s%29sL

Substituting I(s) using the previous equation we derived gives:

http://latex.codecogs.com/gif.latex?V_%7BL%7D%28s%29%20%3D%20%5Cfrac%7BsLV_%7Bin%7D%28s%29%7D%7BsL&plus;R%7D

Our Vin is a Heaviside function, which has a transform of 1/s, so substituting this is gives us:

http://latex.codecogs.com/gif.latex?V_%7BL%7D%28s%29%20%3D%20%5Cfrac%7B1%7D%7Bs%7D%5Ccdot%5Cfrac%7BsL%7D%7BsL&plus;R%7D%20%3D%20%5Cfrac%7BL%7D%7BsL&plus;R%7D

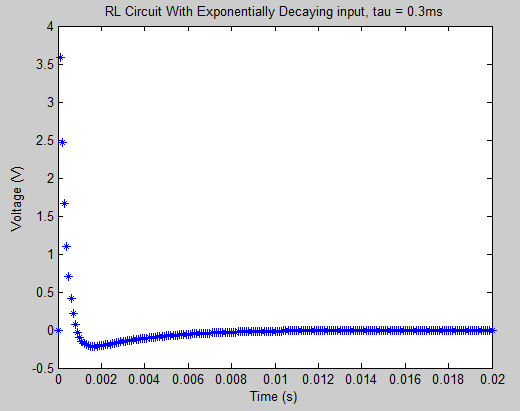
Taking the inverse transform gives:

http://latex.codecogs.com/gif.latex?V_%7BL%7D%28t%29%20%3D%20e%5E%7B-t%5Cfrac%7BR%7D%7BL%7D%7D

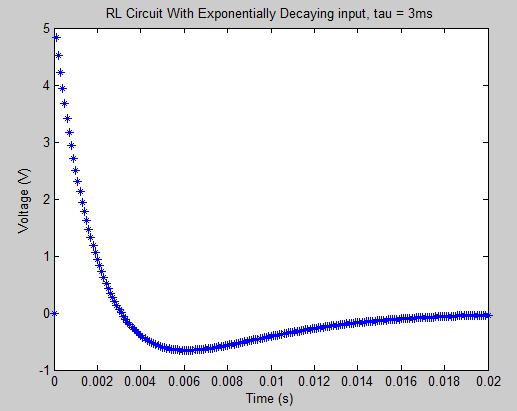
Which explains the exponentially decaying shape of the curve we get.

**Input: Exponential Decay – h = 0.001, tf = 0.02**

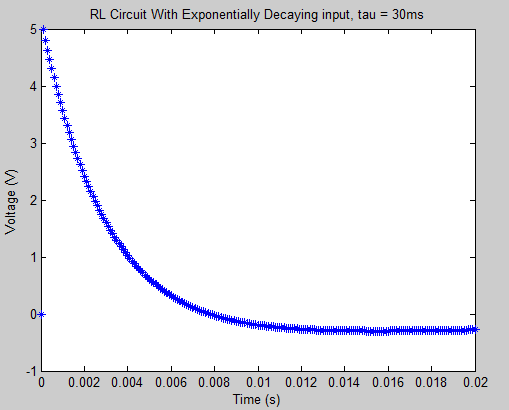
* **τ = 0.3ms**



* **τ = 3ms**



* **τ = 30ms**

****

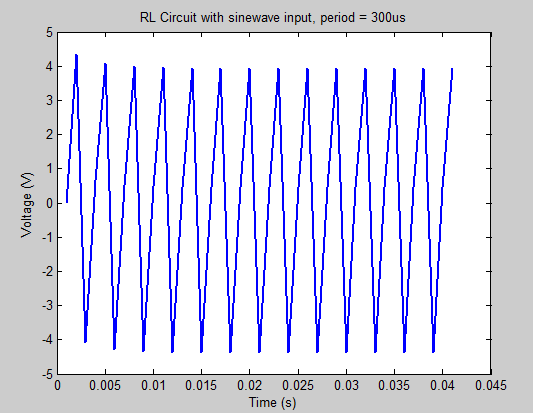
We can see that as τ gets smaller, the response tends towards a horizontal line at y = 0. This is because a small τ value causes the exponent to be very negative, so the exponential term becomes very small and significantly outweighs whatever constant factor it is being multiplied by (in this case 5). The zero input response of this circuit gives zero output, and that is what the graph portrays.

If τ is larger, the absolute value of the exponent becomes small, and the exponential term tends to 1, meaning it is insignificant in comparison to the constant factor, so our input essentially becomes equivalent to the Heaviside input shown earlier, which is why the graphs are similar.

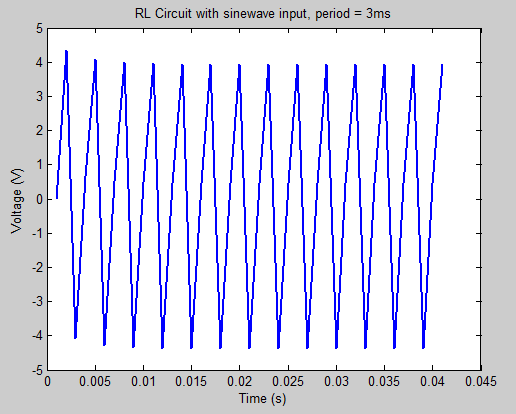
The reason the graph dips below the 0V mark for a short time with exponential input is that the input voltage is decreasing, and a drop in voltage causes a negative change in current. Since the output voltage is proportional to the change in current, the output voltage becomes negative. However, as the exponentially decaying input starts to approach 0, the drop in voltage over time decreases and so the negative change in current becomes less significant, which makes our output graph level out.

**Input: 5V Sine Wave**

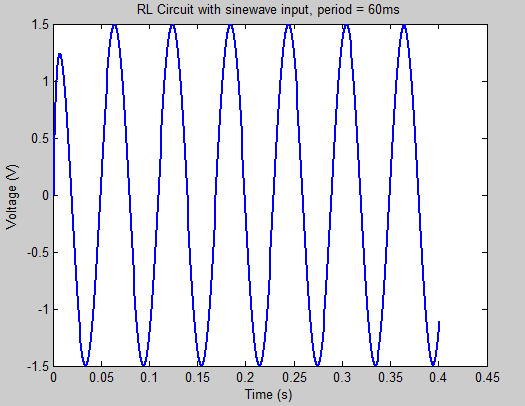
* **T = 300µs, h = 0.001, tf = 0.04**

****

* **T = 3ms, h = 0.001, tf = 0.04**

****

* **T = 60ms, h = 0.001, tf = 0.4**

****

A sinusoidal input to our RL circuit produces a sinusoidal output, as the output is proportional to the rate of change of current, and oscillating change in voltage means that the current will also oscillate.

A low frequency input will have a lower amplitude output because the changes in voltage are less steep, so the response from the inductor will be smaller.

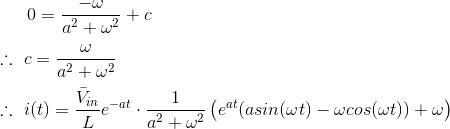
The exact solution can be calculated using the following:

http://latex.codecogs.com/gif.latex?%5Cfrac%7Bd%7D%7Bdt%7Di%28t%29%20&plus;%20ai%28t%29%20%3D%20%5Cfrac%7B%5Cbar%7BV%7D_%7Bin%7D%7D%7BL%7D%20sin%28%5Comega%20t%29http://latex.codecogs.com/gif.latex?a%20%3D%20%5Cfrac%20%7BR%7D%7BL%7D%2C%20%5Comega%20%3D%20%5Cfrac%20%7B2%20%5Cpi%7D%7BT%7D

http://latex.codecogs.com/gif.latex?%5Cfrac%7Bd%7D%7Bdt%7Di%28t%29%20&plus;%20%5Cfrac%7BR%7D%7BL%7Di%28t%29%20%3D%20%5Cfrac%7B%5Cbar%7BV%7D_%7Bin%7D%7D%7BL%7D%20sin%28%5Cfrac%20%7B2%20%5Cpi%7D%7BT%7D%20t%29

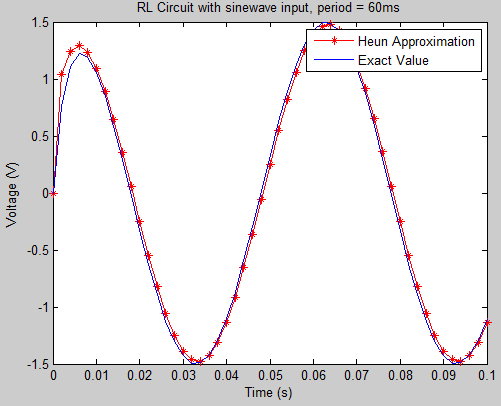
http://latex.codecogs.com/gif.latex?%5Cmu%20%28t%29%3D%20e%5E%7B%5Cint%20a%5C%20dt%7D%20%3D%20e%20%5E%7Bat%7D****http://latex.codecogs.com/gif.latex?P%28t%29%20%3D%20a%2C%5C%20Q%28t%29%20%3D%20%5Cfrac%7B%5Cbar%7BV%7D_%7Bin%7D%7D%7BL%7Dsin%28%5Comega%20t%29Now we solve this first order ODE:

Since i(0) = 0:

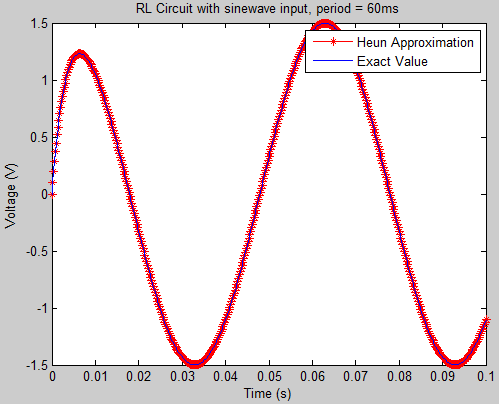
 ****

**** Substituting this into our equation for Vout finally gives us:

Plotting our Heun approximation against the exact solution with a step size of *0.002* yielded this graph:

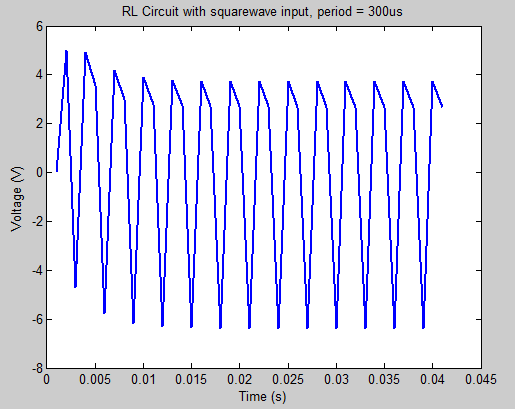


As you can see, there is a lot of error between the approximation and the exact solution. Decreasing the step size to *0.0002* yields a much better approximation:

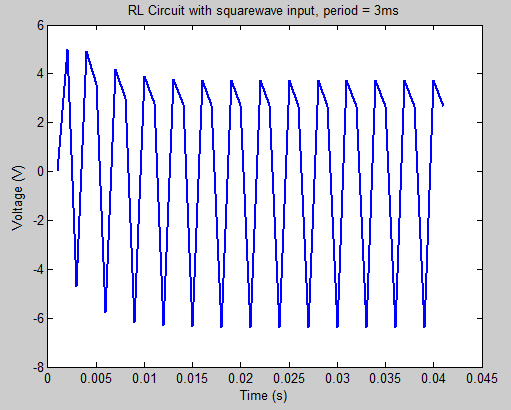


**Input: 5V Square Wave**

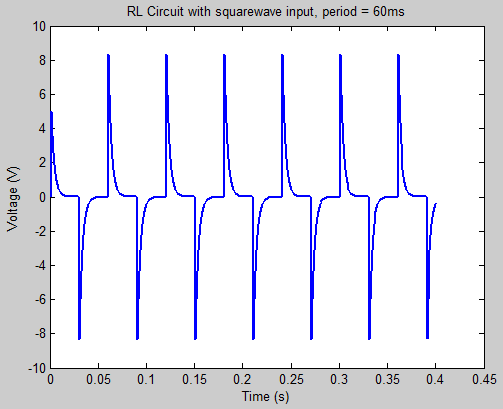
* **T = 300us, h = 0.001, tf = 0.04**

****

* **T = 3ms, h = 0.001, tf = 0.04**

****

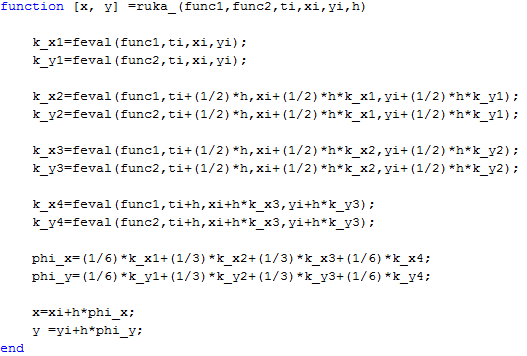
* **T = 60ms, h = 0.001, tf = 0.4**

****

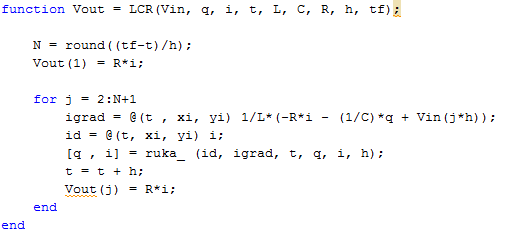
Looking at the T=60ms response, we see that the circuit treats the input as several Heaviside inputs, with opposing signs. When the frequency is higher, the system does not have time to reach a steady state by the time the next edge occurs, so the output oscillates a lot more.

1. **RLC Circuit**

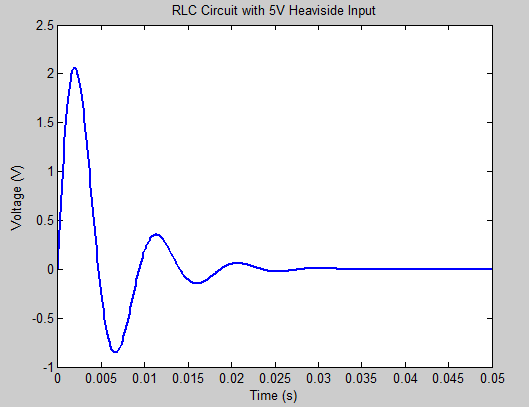
Script ruka\_.m that implements the 4th-order Runge-Kutte



Script LCR.m that implements the 4th-order Runge-Kutte and the equation of the system:

****

**Input: 5V Heaviside:**

****

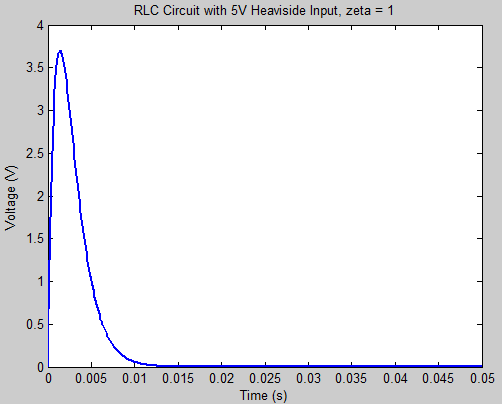
This graph shows the output with all the given input parameters. The signal fades with a damping factor given by:

http://latex.codecogs.com/gif.latex?%5Czeta%20%3D%20%5Cfrac%7BR%7D%7B2%7D%5Csqrt%7B%5Cfrac%7BC%7D%7BL%7D%7D

, which in the case of the given values is 0.3019. Changing R to be equal to

http://latex.codecogs.com/gif.latex?R%20%3D%20%5Cfrac%7B2%7D%7B%5Csqrt%7B%5Cfrac%7BC%7D%7BL%7D%7D%7D

Causes zeta to become 1, and produces the following graph which demonstrates perfect damping:



**Input: 5V Sinusoid:**