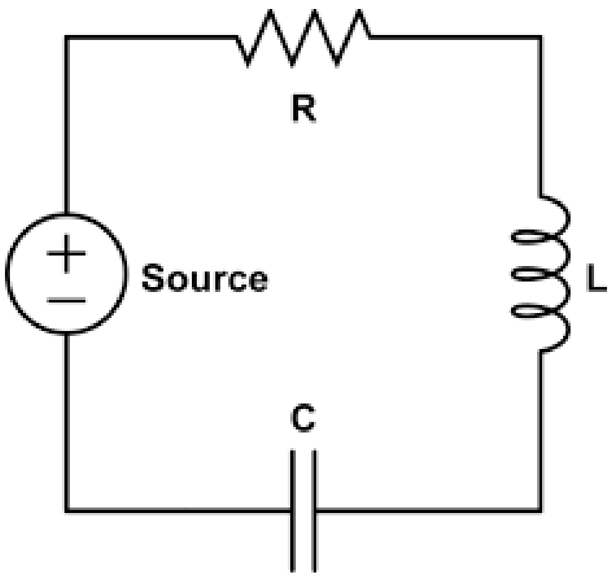
EM Oscillations in an LCR Ciruit

1 Introduction

An LCR circuit is an electrical circuit consisting of an inductor (L), a capacitor (C), and a resistor(R), connected in series or parallel. The RLC circuit is an electrical analog of a spring-mass system with damping.

Considering a series RLC circuit with a constant driving electro-motive force (emf) E,



According to **Kirchoff's law**, the sum of the voltage drops in a closed RLC circuit equals the applied voltage and the current equation for the circuit is:

$$L\frac{di}{dt} + Ri + \frac{1}{C}\int idt = E \tag{1}$$

Differentiating both sides w.r.t time,

$$L\frac{d^2i}{dt^2} + R\frac{di}{dt} + \frac{1}{C} = 0 \tag{2}$$

Differentiating both sides w.r.t time,

$$L\frac{d^2i}{dt^2} + R\frac{di}{dt} + \frac{1}{C} = 0 \tag{3}$$

This is a second-order linear homogeneous differential equation where R is resistance (ohms), L is inductance (henry) and C is capacitance (farads).
Dividing equation (2) by L , we get:

$$\frac{d^2i}{dt^2} + \frac{R}{L}\frac{di}{dt} + \frac{1}{LC} = 0 \tag{4}$$

To solve the differential equation (3) using analytical method, we start with its corresponding **auxiliary equation**:

$$n^2 + \frac{R}{L}n + \frac{1}{LC} = 0 \tag{5}$$

Using **quadratic formula**, the roots are:

$$n_1 = \frac{-R}{2L} + \frac{\sqrt{R^2 - 4(L/C)}}{2L} \tag{6}$$

$$n_2 = \frac{-R}{2L} - \frac{\sqrt{R^2 - 4(L/C)}}{2L} \tag{7}$$

Defining two important parameters,

- **Resonant Frequency-**
  It is defined as the frequency at which the circuit has minimum impedance.

$$f = \frac{1}{\sqrt{LC}} \tag{8}$$

- **Damping Coefficient-**
  It is defined as the amount by which the oscillations of a circuit gradually decrease over time. It is represented as:

$$\alpha = \frac{R}{2L} \tag{9}$$

2 Parameters Required for the C++ Code

To simulate the oscillations produced in the LCR Circuit, we need the following parameters:

Parameters Input by User	Symbol
Inductance	L
Capacitance	C
Resistance	R
Initial charge	Q
Initial Current	I
Final Time	finalTime
Number of iterations	n

Table 1

The step size (h) is computed using the formula:

$$h = (finalTime - 0.0)/n \tag{10}$$

3 EM Oscillations

Three types of oscillations are:

- **Over-damped Oscillations:**

$$R^2 > \frac{4L}{C} \tag{11}$$

  Both the roots, a and b of equation (4) are real and distinct, hence the general solution of the DE (3) is given by,

$$i(t) = Ae^{at} + Be^{bt} \tag{12}$$

- **Critically Damped Oscillations:**

$$R^2 = \frac{4L}{C} \tag{13}$$

  Both the roots of the equation (4) are real and equal; hence, the general solution of the DE (3) is given by,

$$i(t) = (A + Bt)e^{\frac{-R}{2L}t} \tag{14}$$

- **Under-Damped Oscillations:**

$$R^2 < \frac{4L}{C} \tag{15}$$

Both the roots of equation (4) are complex, hence the general solution of the DE (3) is given by,

$$i(t) = \exp(r_p t)(A \cos im_p t + B \sin im_p t) \tag{16}$$

where,

$$r_p = realpart \tag{17}$$

and

$$im_p = imaginarypart \tag{18}$$

4 Algorithm

1. **Include all the required header files.**
2. **Define a class 'LCR'** which has all the above-listed parameters in Table 1 and pre-defined parameters declared as private members.
3. **Declare public member functions** in the class that can be accessed from outside the class using an object or instance of the class defined in the main function (the compiler starts the execution of any C++ program from the main() function)
4. **Create an object of class** inside the class itself. This is another way of creating the object of a class, in some other problems, the object is created in the main() function.
5. **Definition of Member function 1:** LCR(){//}
   The constructor initializes the parameters, including inductance, capacitance, resistance, initial charge, initial current, and final time.
   - Return-type: none
     A constructor is a special member function having the same name as the class and it gets invoked automatically as the instance/object of the class is declared.
   - The function doesn't accept any parameters.
     The function performs certain practical conversions.
     - The input for inductance is in milli henry, which is converted to henry.
     - The input for capacitance is in micro-farad, which is converted to farad.
     - The initial charge is converted from milli-coulombs to coulombs.
     - The initial current is converted from milliamperes to amperes.
     - The final time is converted from microseconds to seconds.
     - The time step `h` is calculated based on the total time and the number of iterations specified by the user.
6. **Definition of Member Function 2:** double derivative (double Q, double I) {//}
   This function calculates the first-time derivative of current (eq. 22) at each time step using the following equations:
   We know that

$$i = \frac{dq}{dt} \tag{19}$$

   Equation (1) can also be written as:

$$L\frac{d^2q}{dt^2} + R\frac{dq}{dt} + \frac{q}{C} = 0 \tag{20}$$

   This gives,

$$L\frac{di}{dt} + Ri + \frac{q}{C} = 0 \tag{21}$$

   Finally, we get

$$\frac{di}{dt} = -\frac{Ri + \frac{q}{C}}{L} \tag{22}$$

- Return type: `double`
  The function returns the calculated value of the current derivative to the calc() function from where the function call has been made.
- The function accepts two parameters
  - `Q`: The charge in the circuit at the current timestep.
  - `I`: The current in the circuit at the current timestep.

7. **Definition of Member Function 3**: void condition(void) {//}
   This function checks the nature of the oscillations in the LCR circuit based on the relationship between resistance, inductance, and capacitance as given in the equations (11), (13), and (15) above.
   - Return type: `void` The function does not return any value.
   - The function does not accept any parameters.
   - The function uses the if-else-if condition to check the type of oscillations produced in the LCR circuit.

```
if(condition 1)
{
    \\ execute this code block if condition 1 is true
}
else if(condition 2)
{
    \\ execute this code block if condition 2 is true
}
else
{
    \\ execute this code block if condition 1 is false and condition 2 is false
}
```

8. **Definition of Member Function 4:** `void calc()` This function implements the Runge-Kutta method of order 4 using a 'for-loop' to calculate the charge and current in the circuit at each time step.
   - Return type: `void`
     The function does not return any value.
   - The function does not accept any parameter.
   - Following is the list of operations performed by this function:
     - The function makes the function call to the derivative (ddouble, double) function to implement the RK4 method. (Check more about RK4 from the documentation on Runge_Kutta Method)
     - The function enters the value of charge and current computed at each time step by 'for-loop' into the data file.
9. **Define the main function:** `int main()`
   - The instance/object created inside the 'LCR' class makes function calls to all necessary functions.
   - Return 0 to indicate successful execution.
10. The final step in the algorithm involves a crucial step i.e., plotting the data saved in the file by the calc() function using **Gnuplot software**.
    Gnuplot is a graphing utility for Linux, OS/2, MS Windows, OSX, VMS, and many other platforms. Its source code is freely distributed and is extensively used for data visualization.
    Follow the following steps to plot data on Gnuplot:
    (a) Download Gnuplot.
    (b) Go to → Change directory.
    Select the folder where the 'lcr.txt' data file is stored.

(c) Write the following command in Gnuplot:

```
plot "lcr.txt" u 1:2 w l \\ and
plot "lcr.txt" u 1:3 w l
\* This command asks the Gnuplot to plot Column 2 with respect to Column 1
    with a line-type graph and plot Column 3 with respect to Column 1 with a
    line-type graph. There are many other variations to this command, you can
    check them out at the official website of Gnuplot or watch some free
    tutorials on YouTube.*/
```