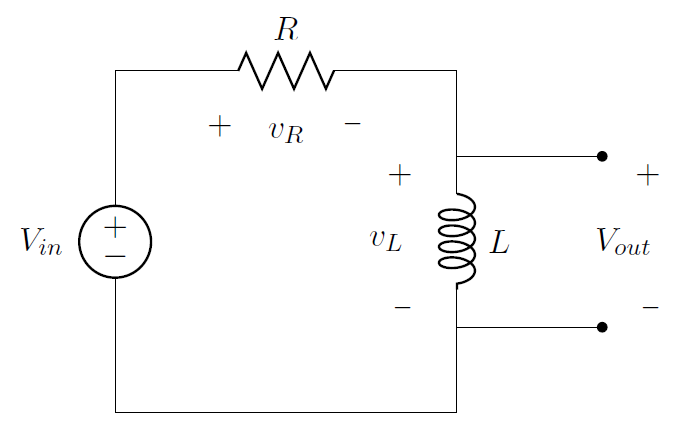
**General Method for Solving LR Circuit**

The task is to numerically derive the output response of an LR high pass filter as shown in figure ??. The voltages and currents in such a circuit are relates by equations ?????.

Given that the voltage across the inductor is proportional to the inductor current, with constant of proportionality L, the output voltage, which is equivalent to the inductor voltage can be derived indirectly from the rate in change of current in the circuit . Note that current in the inductor and the resistor are equal. By rearranging equation ??? we can define from current and voltage in as follows in equation ????.

equation ???? can be simply solved using numerical analysis methods for first order equations. Once rate of change current is calculated, output voltage is simply derived by multiplication with the inductance of the inductor. In particular, this section will look at the use of second order Runge-Kutta methods to solve this equation.



**The Ralston method**

The Ralston method is a second order Runge-Kutta used to solve first order differential equations.

The method was implemented for an LR low pass filter as in figure ?? where,

* VinFunc is a voltage input function,
* R is the resistor value
* L is the inductor inductance
* i0 is the current at time = 0s
* tf is the end time for the numerical analysis
* n is the number of samples



The local function **didt** (line 8) defines the circuit equation (DEFINE SOMEWHERE?). For a given time and current value this local function will return the value of . The variable **h** (line 13) is initialized to hold the interval width between samples given the number of samples, **n**, and the end time, **tf**. A temporary variable **tmpi** (line 14) holds the last computed value of current. This is initialized to the value of the input parameter **i0** to give the starting condition for the circuit. On successive iterations of the **for** loop (line 17-22) **tmpi** is overwritten and used as a shift register to supply the value of current for the next iteration. The array **iDelta** (line 15) is declared to be used as a store for values of at intervals defined by **h**.