[Getting Physical](http://book.xogeny.com/behavior/equations/physical/)

[A Mechanical Example](http://book.xogeny.com/behavior/equations/mechanical/)



[http://book.xogeny.com/_static/images/TitleHeading.png](http://book.xogeny.com/)

Release: v0.3.6-1-gc28d7f1-Early Access

[Basic Equations](http://book.xogeny.com/behavior/equations/)

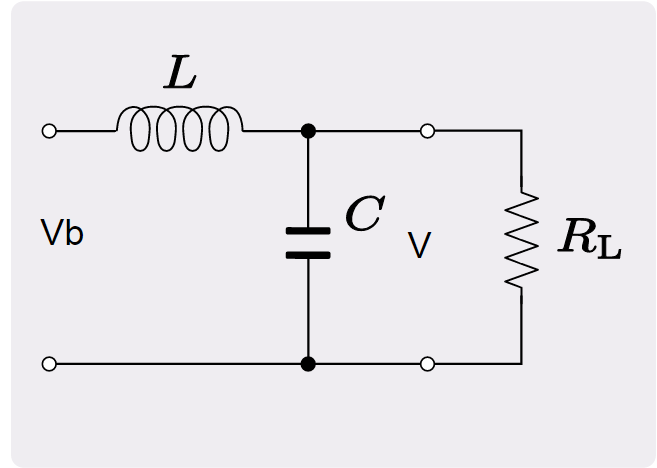
**An Electrical Example**

Table Of Contents

* [Basic Equations](http://book.xogeny.com/behavior/equations/)
  + [An Electrical Example](http://book.xogeny.com/behavior/equations/electrical/)

An Electrical Example

Let us return now to an engineering context. For readers who are more familiar with electrical systems, consider the following circuit:



Low-Pass RLC Filter

Suppose we want to solve for: V, iL, iR and iC. To solve for each of the currents iL, iR and iC, we can use the equations associated with inductors, resistors and capacitors, respectively:

V=iRR

CdV/dt=iC

LdiL/dt=(Vb−V)

where Vb is the battery voltage.

Since we have only 3 equations, but 4 variables, we need one additional equation. That additional equation is going to be Kirchoff’s current law:

iL=iR+iC

Now that we have determined the equations and variables for this problem, we will create a basic model (including physical types) by translating the equations directly into Modelica. But in a later section on [*Electrical Components*](http://book.xogeny.com/components/components/elec_comps/#electrical-components) we will return to this same circuit and demonstrate how to create models by dragging, dropping and connecting models that really look like the circuit components in our [*Low-Pass RLC Filter*](http://book.xogeny.com/behavior/equations/electrical/#low-pass-rlc).

But for now, we will build a model composed simply of variables and equations. Such a model could be written as follows:

**model** **RLC1** "A resistor-inductor-capacitor circuit model"

**type** **Voltage**=Real(unit="V");

**type** **Current**=Real(unit="A");

**type** **Resistance**=Real(unit="Ohm");

**type** **Capacitance**=Real(unit="F");

**type** **Inductance**=Real(unit="H");

**parameter** Voltage Vb=24 "Battery voltage";

**parameter** Inductance L = 1;

**parameter** Resistance R = 100;

**parameter** Capacitance C = 1e-3;

Voltage V;

Current i\_L;

Current i\_R;

Current i\_C;

**equation**

V = i\_R\*R;

C\*der(V) = i\_C;

L\*der(i\_L) = (Vb-V);

i\_L=i\_R+i\_C;

**end** **RLC1**;

Let’s go through this example bit by bit and reinforce the meaning of the various statements. Let’s start at the top:

**model** **RLC1** "A resistor-inductor-capacitor circuit model"

Here we see that the name of the model is RLC1. Furthermore, a description of this model has been included, namely "A resistor-inductor-capacitorcircuit model". Next, we introduce a few physical types that we will need:

**type** **Voltage**=Real(unit="V");

**type** **Current**=Real(unit="A");

**type** **Resistance**=Real(unit="Ohm");

**type** **Capacitance**=Real(unit="F");

**type** **Inductance**=Real(unit="H");

Each of these lines introduces a physical type that specializes the built-in Real type by associating it with a particular physical unit. Then, we declare all of the parameter variables in our problem:

**parameter** Voltage Vb=24 "Battery voltage";

**parameter** Inductance L = 1;

**parameter** Resistance R = 100;

**parameter** Capacitance C = 1e-3;

These parameter variables represent various physical characteristics (in this case, voltage, inductance, resistance and capacitance, respectively). The last variables we need to define are the ones we wish to solve for, *i.e.*,

Voltage V;

Current i\_L;

Current i\_R;

Current i\_C;

Now that all the variables have been declared, we add an equation section to the model that specifies the equations to use when generating solutions for this model:

**equation**

V = i\_R\*R;

C\*der(V) = i\_C;

L\*der(i\_L) = (Vb-V);

i\_L=i\_R+i\_C;

Finally, we close the model by creating an end statement that includes the model name (*i.e.*, RLC1 in this case):

**end** **RLC1**;

One thing that distinguishes this example from the previous examples is the fact that it contains more equations. As with the NewtonCooling example, we have some equations with expressions on both the left and right hand sides. We also have a mix of differential equation (ones that include the derivative of a variable) and others that are simply algebraic equations.

This further emphasizes the point that in Modelica it is not necessary to put the system of equations into the so-called “explicit state-space form” required in some modeling environments. We could, of course, rearrange the equations into a more explicit form like this:

der(V) = i\_C/C;

der(i\_L) = (Vb-V)/L;

i\_R = i\_L-i\_C;

V = i\_R\*R;

But the important point is that with Modelica, we do not need to perform such manipulations. Instead, we are free to write the equations in whatever form we chose.

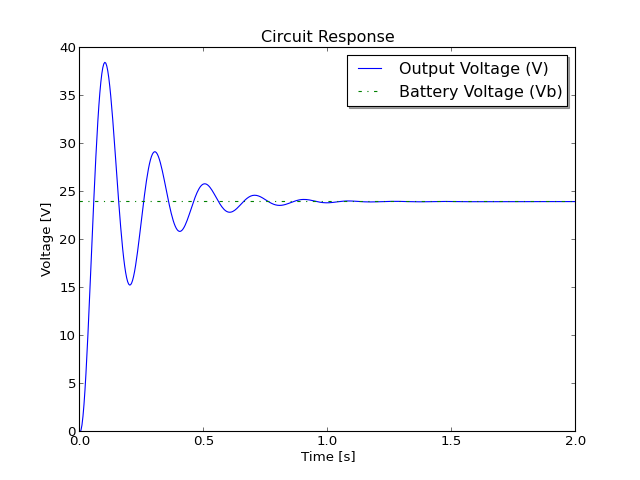
Ultimately, these equations will probably need to be manipulated into a form like explicit state-space form. But if such manipulations are necessary, it will be the responsibility of the Modelica compiler, not the model developer, to perform these manipulations. This eliminates the need for the model developer to deal with this tedious, time consuming and error prone task.

The ability to keep equations in their “textbook form” is important because, as we will show in later sections, we eventually want to get to the point where these equations are “captured” in individual components models. In those cases, we won’t know (when we create the component model) exactly what variable each equation will be used to solve for. Making such manipulations the responsibility of the Modelica compiler not only makes the model development faster and easier, but it dramatically improves the **reusability** of the models.

The following figure shows the dynamic response of the RLC1 model:

([Source code](http://book.xogeny.com/plots/RLC1.py))

 Simulate RLC1 in your browser



**Expanding on these electrical examples**

As mentioned in the [*Preface*](http://book.xogeny.com/front/preface/#preface), the structure of this book allows us to explore a more hypermedia based approach in which readers are encouraged to process the material that is most aligned with their goals and interests. The next chapter will present a model whose equations are derived from a mechanical system. If you would prefer instead to see this electrical example extended to include more complex behavior, you may want to skip ahead to the [*Switched RLC Circuit*](http://book.xogeny.com/behavior/discrete/switching/#switched-rlc) example.