Solving ordinary differential equations using MATLAB

Introduction

MATLAB offers a range of 'solvers' for ordinary differential equations. In this section we will concentrate on using MATLAB's basic method for solving differential equations of the type

$$\frac{dy}{dt} = f(t, y) \tag{1}$$

This form includes the equation for a first order system, namely

$$\frac{dy}{dt} = -ay + b(t) \tag{2}$$

In MATLAB, the basic solver is called **ode45**. '**ode**', of course, stands for ordinary differential equation; the **45** bit is a cryptic reference to the method of solution which dates from 1980 (see MATLAB help if you want to know more). For us, **ode45** is a tool and we are grateful to the mathematicians who provided it!

MATLAB does not deliver an algebraic functional form for the solution y(t). Since it can only calculate numbers, it presents the solution as a set of values of y(t) at times which the user (you) specifies. The set of values is presented as a **column vector**, Y, of the form

$$Y = \begin{bmatrix} y(t_1) \\ y(t_2) \\ y(t_3) \\ y(t_4) \\ . \\ . \\ . \end{bmatrix} \tag{3}$$

To see what the solution y(t) looks like you have to plot the vector Y.

Programming MATLAB to solve an ode

What information does MATLAB need to solve the ode?

To specify the task completely we need to tell MATLAB

(i) the functional form of the ode, i.e. specify f(t,y) in equation (1).

(ii) the range of 't' for which the solution is required.

(iii) and, finally, so that MATLAB can evaluate the constant of integration, we need to include a boundary condition, in this case the initial value of y(t).

The *syntax* in MATLAB for the solver **ode45** is:

[t , Y] = ode45(@rhsf, *tspan*, y0)

The elements in ode45( ) relate to (i), (ii) and (iii) above as follows:

(i) @rhsf

@ is just a symbol required by MATLAB to precede the name of the function, which I have arbitrarily called 'rhsf' for 'right-hand-side function'. It can be given any name.

The solver ode45 assumes that the ode has the form of equation (1) and only needs to be given the explicit form of the function on the right hand side. The **function** is specified in the usual way (see example below).

MATLAB will call this function a great many times as it works iteratively through the problem.

(ii) *tspan*

tspan is the range of the independent parameter, t. It is specified as a **row vector**. It can be specified in the usual way:

[*start value* **:** *step* **:** *finish value*]

or, MATLAB can be left to decide the best step value, in which case *tspan* is simply

[*start value* *finish value*] (two numbers separated by a space)

(iii) y0

This is the value of y(t) at t = *start value* (normally t = 0)

The output: [t , Y]

The output of the solver contains the two **vectors** needed to plot the solution**.**

The **vector,** t, is *tspan, i.e.* the set of time values at which y is calculated, say t_1, t_2, t_3, etc.

The **column vector,** Y, is the set of solution values $y(t_1)$, $y(t_2)$, $y(t_3)$ …etc. as indicated in equation (3).

A graph of the solution can be generated by the command:

```
plot(t,Y)
```

Example:

We take as an example the equation governing the voltage, V_1, on the capacitor in an RC circuit when the circuit is driven by an alternating voltage $V_{bo}\sin(\omega t)$. The details are given in Appendix A.

The equation is the familiar first order one:

$$\frac{dV_1}{dt} = \left(\frac{V_{bo}}{\tau}\right)\sin(\omega t) - \left(\frac{1}{\tau}\right)V_1 \qquad (4)$$

This has the form of equation (2).

Now we should learn another trick.

The right-hand-side function is:

$$\left(\frac{V_{bo}}{\tau}\right)\sin(\omega t) - \left(\frac{1}{\tau}\right)V_1$$

To evaluate this, the function needs the values of τ, V_{bo} and ω, in addition to the dependent and independent variables V_1 and t.

These extra parameters could be allocated values within the function M-file itself; but, if they are to be changed, it is easier to define them in the main solver M-file.

The 'function' M-file

So the M-file for the rhsf is:

[Note that the function name in this example is f3 and MUST be saved as f3.m.]

```
function q=f3(t,V1,tau,Vb0,omega)
q= (Vb0./tau).*sin(omega.*t)-V1./tau;
```

Now we need to make some decisions about the parameters to use in the solution.

Choosing parameters

Vbo, τ and ω

We know that the amplitude of the 'drive', V_{bo}, only serves to make the amplitude of the response, V_1, bigger or smaller. So we will arbitrarily choose

$$V_{bo} = 5 \text{ volts}$$

More interest attaches to roles of ω and τ in determining V_1. We know that the output will be noticeably reduced and will lag significantly in phase if $\omega\tau > 1$. So to make things interesting, we will choose

$$\omega = 10^6 \text{ rad/s} \qquad (f = \omega/2\pi \approx 160 \text{ kHz})$$

and $$\tau = 4\times10^{-6} \text{ s}$$

so $$\omega\tau = 4$$

tspan

We will aim to show a plot of the first 10 cycles of the evolution. Since the frequency is about 160 kHz, the period of oscillation is about $T = 6\times10^{-6}$s. So, in the first instance the integration will be over the interval:

$$t = 0 \text{ to } 60\times10^{-6}\text{s.}$$

We will look at the evolution from 'switch-on'. The applied voltage, $V_{bo}\sin(\omega t)$, is 0 at $t = 0$ and V_1 is also zero at the start.

The solver M-file

Make sure that you understand the commands in the program.

```
V10=0;                      % the starting value of V1, called V10, is zero.
tau=4e-6;                   % 4e–6 means 4×10^–6.
Vb0=5;
omega=1e6;
T10=10*2*pi/omega;          % this is the duration of 10 cycles.
t=[0:T10/1000:T10];         % the solver will produce 1000 points for the 10 cycles.
[t,V1]=ode45(@f3,t,V10,[],tau,Vb0,omega);
                            % NOTE the protocol ',[],' before the passed parameters.
Vb=Vb0.*sin(omega.*t);      % the drive voltage which will also be plotted.
figure(1)
plot(t,V1,t,Vb)
```

Exercises

(2.1) Run the foregoing M-file program to show the nature of the response for values of $\omega\tau < 1$, $= 1$ and > 1. Satisfy yourself that the behaviour of the output amplitude and phase is in accordance with the analytical steady state solutions derived in class.

Note the transient behaviour between start-up and the attainment of the steady state.

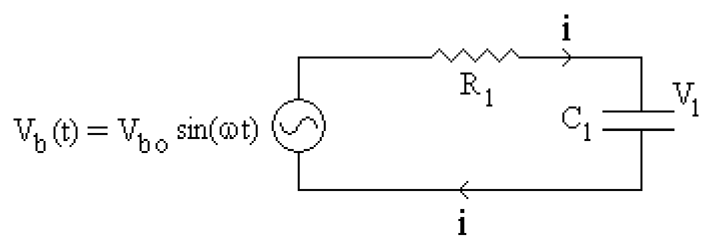
(2.2) Plot the response, from 0 to 5τ, of the same system to a step input of $V_b = 5$ V.

Appendix

The RC circuit driven by an alternating voltage

We will use Matlab to solve the problem that we solved with the help of complex numbers.

The set-up for the RC circuits is as shown.



The alternating voltage $V(t) = V_{bo}\sin(\omega t)$ drives current to and fro round the circuit.

The question is: "How does V_1 evolve with time?"

We can set up the differential equation for V_1 starting from the usual circuit equations.

$$i = \frac{V_b - V_1}{R_1} \qquad\qquad i = \frac{dQ_1}{dt} = C_1\frac{dV_1}{dt}$$

From these

$$\frac{dV_1}{dt} = \frac{i_1}{C_1} = \frac{V_b - V_1}{R_1C_1} \qquad \text{(i)}$$

We will rewrite (i) by putting $\tau = R_1C_1$

It becomes

$$\frac{dV_1}{dt} = \frac{V_b - V_1}{\tau} \qquad \text{(ii)}$$

or

$$\frac{dV_1}{dt} = \frac{V_{bo}}{\tau}\sin(\omega t) - \frac{V_1}{\tau}$$

[equation (4) of main text]