1 Maple and Differential Equations

1.1 Introduction

The basic Maple command for solving scalar differential equations is `dsolve`. This command can be used to obtain analytical solutions of linear equations as well as numerical solutions of nonlinear equations. The basic syntax of the `dsolve` command for a single linear equation is

```maple
> dsolve({deq, initcond},func);
```

where `deq` represents a differential equation, `initcond` stands for one or more initial conditions, and `func` is the function representing the solution of the differential equation.

Several other Maple commands will be useful to us, including the `diff` command for computing derivatives, the `subs` command for substituting sets of parameter values into differential equations and their solutions, and several commands from the `DEtools` package for plotting slope fields and phase portraits.

This document provides examples of for all of these commands. It begins with examples of solving linear first and second order differential equations and then goes on to describe the plotting commands from the `DEtools` package.

1.2 Linear first-order equations

Consider the following differential equation.

\[
\frac{dx}{dt} = -kx
\]

The solution, obtained by separation of variables (or otherwise) is

\[
x(t) = C \exp(-kt),
\]

where \(C\) is a constant. The Maple command to solve this differential equation is given below, as well as the output.

```maple
> dsolve(diff(x(t),t) = -k*x(t) ,x(t));
```

\[
x(t) = e^{(-kt)}C1
\]

The first thing to notice is that the Maple command needs you to specify explicitly in a couple of places that \(x\) depends on \(t\). We don’t always do this when writing down differential equations by hand, but Maple often needs more guidance.

Actually Maple isn’t quite that stupid, because the following shortcut does work, though recent versions of Maple give a warning if you do this.

```maple
> dsolve(diff(x(t),t) = -k*x ,x(t));
```

\[
x(t) = e^{(-kt)}C1
\]
However, if you replace either of the remaining occurrences of \( x(t) \) with \( x \) you will get an error. To avoid problems, I would stick with using \( x(t) \) all the time. You can never go wrong by underestimating the intelligence of Maple!

Another thing to notice is that Maple uses \( _C1 \) to stand for the constant \( C \). Putting the underscore in front is just Maple’s way of minimizing the risk of a conflict with a variable you have set. You might try to see what Maple does if you give the variable \( _C1 \) a value.

The final thing to notice is that the output of the \texttt{dsolve} command is an equation, with the expression for the solution appearing on the right-hand side of the equation. If you need this expression for plotting or further manipulation, you will have to somehow extract it from the equation. If you are using \texttt{dsolve} to solve a single differential equation, then using the \texttt{rhs} command is the easiest way to accomplish this.

To use the \texttt{rhs} command, you have to be able to access the output of the \texttt{dsolve} command. The best way to do this is to give the \texttt{dsolve} command a label, as in the following example. It usually makes sense to also provide a label for the expression you’ve extracted, so it will be easy to plot it, differentiate it, or otherwise use it.

\[
> \texttt{sol1 := dsolve(diff(x(t),t) = -k*x(t) ,x(t));}
\]
\[
\texttt{sol1 := x(t) = e^(-kt) \_C1}
\]

\[
> \texttt{x1 := rhs(sol1)}
\]
\[
\texttt{x1 := e^(-kt) \_C1}
\]

You can also specify initial conditions in the \texttt{dsolve} command. For example, we could solve the IVP

\[
\frac{dx}{dt} = -kx \quad x(0) = 120
\]

with the following command.

\[
> \texttt{sol2 := dsolve\{diff(x(t),t) = -k*x(t),x(0) = 120\},x(t)}
\]
\[
\texttt{sol2 := x(t) = 120 e^(-kt)}
\]

Note that you have to put the differential equation and the initial condition together and enclose them in curly braces (\{\}).

The simple example we have been using includes a parameter \( k \). Before we can plot the solution we have to set its value. If you are going to set one value for \( k \) and not change it, then you can just use a command like the first one below. The other command just shows that the value of \( k \) has been set.

\[
> \texttt{k := 1/4;}
\]
\[
\texttt{k := 1 \_4}
\]

\[
> \texttt{x1;}
\]
\[
\texttt{e^(-1/4t) \_C1}
\]
However, sometimes you want to give a parameter several values, perhaps to plot the solution for different parameter values or because you need the general form of the solution for something else. In these cases, setting the values of parameters is best done with the `subs` command, as shown below. (The first command just unsets the value of \( k \) so it is treated as a variable again.)

\[
\begin{align*}
> & \quad k := 'k'; \\
& \quad k := k \\
> & \quad \text{subs}(k=1/4,x1); \\
& \quad e(-1/4t) \_C1
\end{align*}
\]

The final example in this section demonstrates how to plot the solution to the IVP

\[
\frac{dx}{dt} = -kx \quad x(0) = 120
\]

which we labeled above with the name `sol2`. For the plot command to work, Maple has to be able to evaluate the expression being plotted to a number. This means that the value of \( k \) has to be set. The command below shows how to use the `subs` command to do this.

\[
\begin{align*}
> & \quad \text{plot}((\text{subs}(k=1/4,\text{rhs(sol2)})),t=0..5);
\end{align*}
\]

### 1.3 Linear second order equations with constant coefficients

For this section we consider the following second order differential equation with constant coefficients

\[
m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = 0,
\]

where \( m, c \) and \( k \) are constants. The procedure for using Maple to solve this second order equation is very similar to what we did in the previous section, but there are two main differences.

- The differential equation has more than one parameter. This means that set notation will be necessary if the `subs` command is used to set parameter values.

- Initial conditions will involve values of the first derivative of \( x \) as well as values of \( x \). The Maple syntax for specifying values of derivatives is not what you might expect; the `diff` command cannot be used. Instead we will have to use the Maple derivative operator, \( D \).

To proceed, we first define our differential equation and give it a label. This will save some typing later on.

\[
\begin{align*}
> & \quad \text{de1} := m*\text{diff}(x(t),t,t)+c*\text{diff}(x(t),t) + k*x(t) = 0; \\
& \quad \text{de1} := m \left( \frac{\partial^2}{\partial t^2} x(t) \right) + c \left( \frac{\partial}{\partial t} x(t) \right) + k \_x(t) = 0
\end{align*}
\]
Next, we define a set of values for our parameters.

\[ \text{par1} := \{m=1, \ c= \frac{1}{10}, \ k=4\}; \]

Finally, we substitute the parameter values and solve the IVP

\[ \frac{d^2x}{dt^2} + \frac{1}{10} \frac{dx}{dt} + 4x = 0, \ x(0) = 1, \ x'(0) = 0 \]

with the command

\[ \text{sol3} := \text{dsolve} (\text{subs(par1,}\{\text{de1}}, D(x)(0) = 0, x(0) = 1\}, x(t)); \]

\[ \text{sol3} := x(t) = e^{-\frac{1}{20} t} \cos \left( \frac{1}{20} \sqrt{1599} t \right) + \frac{1}{1599} \sqrt{1599} e^{(-1/20 t)} \sin \left( \frac{1}{20} \sqrt{1599} t \right) \]

Note how the initial condition for \( x' \) is specified. We can extract the solution the same way we did before, with the command

\[ \text{x3} := \text{rhs(sol3)}; \]

\[ x3 := e^{-\left(\frac{1}{20} t\right)} \cos \left( \frac{1}{20} \sqrt{1599} t \right) + \frac{1}{1599} \sqrt{1599} e^{\left(-\frac{1}{20} t\right)} \sin \left( \frac{1}{20} \sqrt{1599} t \right) \]

This result can now be plotted or otherwise manipulated as desired.

As a final example, note that above we substituted the parameter values in the differential equation and then solved it. You can proceed by first solving and then substituting parameter values, as we did for the first order equation, but substituting and then solving is usually preferable.

To see this, suppose we solve the IVP with new initial conditions \( x(0) = 1 \) and \( x'(0) = 1 \) without substituting parameter values first, as shown below.

\[ \text{sol4} := \text{dsolve} (\{\text{de1}, D(x)(0) = 1, x(0) = 0\}, x(t)); \]

\[ \text{sol4} := x(t) = m e^{-1/2} \left( \frac{(c \sqrt{c^2 - 4 m k})}{m} t \right) \frac{1}{\sqrt{c^2 - 4 m k}} - m e^{-1/2} \left( \frac{(c + \sqrt{c^2 - 4 m k})}{m} t \right) \frac{1}{\sqrt{c^2 - 4 m k}} \]

Here is the result of substituting our parameter values.

\[ \text{subs(par1,sol4)}; \]

\[ x(t) = -\frac{1}{1599} \sqrt{-1599} \sqrt{100} e^{-1/2} \left( \frac{1/10 - 1/100 \sqrt{-1599} \sqrt{100}}{m} t \right) \]

\[ + \frac{1}{1599} \sqrt{-1599} \sqrt{100} e^{-1/2} \left( \frac{1/10 + 1/100 \sqrt{-1599} \sqrt{100}}{m} t \right) \]

If we do things in the opposite order, the result is simpler, as shown below.

\[ \text{sol5} := \text{dsolve} (\text{subs(par1,}\{\text{de1}, D(x)(0) = 1, x(0) = 0\}), x(t)); \]

\[ \text{sol5} := x(t) = \frac{20}{1599} \sqrt{1599} e^{(-1/20 t)} \sin \left( \frac{1}{20} \sqrt{1599} t \right) \]
2 Plotting direction fields and solution curves

2.1 Introduction

In this section we describe Maple commands for plotting direction fields and/or solution curves for a single first order differential equation of the form

\[ \frac{dx}{dt} = f(t, x) \]  

(1)

or a two-dimensional autonomous system of the form

\[ \frac{dx}{dt} = f_1(x, y) \]
\[ \frac{dy}{dt} = f_2(x, y) \]  

(2)

or a two-dimensional non-autonomous system of the form

\[ \frac{dx}{dt} = g_1(t, x, y) \]
\[ \frac{dy}{dt} = g_2(t, x, y) \]  

(3)

There are four Maple commands we will describe here. All are in the DEtools package. We deal here with only the most basic examples for these commands. For example, we restrict our attention to a single first-order differential equation or a two dimensional autonomous or non-autonomous system, but the Maple commands described here will work with both with a a single higher order differential equation or a system of three or more first order differential equations. To learn about more advanced features, consult the help pages for the individual commands. Here is a list of the four commands we will consider, along with brief descriptions. More complete descriptions and examples follow.

dfieldplot This command will plot the direction field for either a single differential equation or a two-dimensional autonomous system.

phaseportrait Plots solution trajectories for a set of initial conditions. Includes the direction field for autonomous systems by default, but its display can be suppressed when desirable, e.g. when using the scene parameter to plot individual components versus \( t \).

DEplot Will plot direction fields in the same manner as dfieldplot. Also does the same things as phaseportrait.

DEplot3d Plots solution curves in three dimensions, (including time), for a two-dimensional non-autonomous system.
The first three commands do similar things. In fact, dfieldplot and phaseportrait are convenience functions with simplified syntax that call the lower level routine DEplot. We will usually use the higher level routines, so they will be described before DEplot. The DEplot3d command is similar in many respects to DEplot, but plots in three dimensions and does not produce direction fields. As mentioned above, these commands will work with systems of differential equations of dimension 3 or higher, but this won’t be described here.

2.2 Examples

The examples in the rest of this handout will deal with the first order differential equation

$$\frac{dx}{dt} = \sin(x^2)$$  \hspace{1cm} (4)

the two dimensional autonomous system

$$\frac{dx}{dt} = y$$

$$\frac{dy}{dt} = -4x$$ \hspace{1cm} (5)

and the two dimensional non-autonomous system

$$\frac{dx}{dt} = y$$ \hspace{1cm} (6)

$$\frac{dy}{dt} = -y - 4x + 2 \sin(3t).$$

2.3 Plotting direction fields: dfieldplot

The dfieldplot routine plots direction fields for a single first order differential equation or a two-dimensional system of differential equations, but cannot be used with a two-dimensional non-autonomous system. The phaseportrait command, which is described in the next section, can be used to plot solution trajectories for all of the three classes of differential equations being considered here.

2.3.1 Direction fields for a single differential equation

An example of using dfieldplot with a single first order equation appears below. Note that the first command loads the DEtools package. It is only necessary to do this once in each session.

\[
\text{> with(DEtools);}
\]

[DEnormal, DEplot, DEplot3d, Dchangevar, PDEchangecoords, PDEplot, autonomous, convertAlg, convertsys, dfieldplot, indicialeq, phaseportrait, reduceOrder, regularsp, translate, untranslate, varparam]
> de1 := diff(x(t),t) = sin(x(t)^2);

\[ de1 := \frac{\partial}{\partial t} x(t) = \sin(x(t)^2) \]

> dfieldplot(de1,[x],t=0..5,x=-2..2);

The first argument to \texttt{dfieldplot} is always the differential equation (or system of differential equations, as we’ll see below). The second argument lists the dependent variable \( x \). This argument must be present. The third argument specifies a range for the independent variable, and must be present. The fourth argument, specifying a range for the dependent variable, is optional but usually needed so that the command plots the \( x \) range you are interested in. There are several other optional arguments, see the help page for more details.

### 2.3.2 Direction fields for two-dimensional autonomous systems

Next, we present an example of \texttt{dfieldplot} usage for a two-dimensional autonomous system. The main difference from our first example is that two differential equations must be specified.

> sys1 := \{diff(x(t),t) = y(t), diff(y(t),t) = -4*x(t)\};

\[ sys1 := \left\{ \frac{\partial}{\partial t} x(t) = y(t), \frac{\partial}{\partial t} y(t) = -4 \cdot x(t) \right\} \]

> dfieldplot(sys1,[x,y],t=0..1,x=-2..2,y=-2..2);

As in our previous example, only the first three arguments are required. However, I can’t really think of a reason for not specifying ranges for \( x \) and \( y \).

### 2.3.3 Optional arguments to \texttt{dfieldplot}

There are several optional arguments to \texttt{dfieldplot}. One of the most useful is the \texttt{arrows} option, which can be used to change the appearance of the arrows. Here is one example.

> dfieldplot(sys1,[x,y],t=0..1,x=-2..2,y=-2..2,arrows=MEDIUM);

Other possible values for the \texttt{arrows} option are \texttt{LARGE} (large arrows), \texttt{LINE} (no arrow-heads), and \texttt{NONE} (no direction field - don’t use this option with \texttt{dfieldplot}). The default value of this optional parameter is \texttt{SMALL}.

### 2.4 Plotting solution curves: \texttt{phaseportrait}

The \texttt{phaseportrait} command is even more useful. What it does, essentially, is to plot solution trajectories for a set of initial conditions that you specify. For a single differential equation or a two-dimensional system of autonomous differential equations, \texttt{phaseportrait} includes the direction field, unless you specify the value \texttt{NONE} for the parameter \texttt{arrows}.

By default, the \texttt{phaseportrait} command plots the solution of an autonomous system as a parametric curve in the \( xy \) plane. However, it is possible to plot individual
components by using the optional `scene` parameter and specifying the value `NONE` for the parameter `arrows`. In the case of an non-autonomous system, the solution curve is also plotted as a parametric curve in the $xy$ plane by default, but the `scene` parameter can be used to select any two variables for plotting. The `DEplot3d` command can be used to obtain a plot of the trajectories in $txy$ space. Examples are given below.

### 2.4.1 Single, first order differential equations

Using the `phaseportrait` command to plot trajectories for a single first order differential equation is straightforward. See the following example.

```latex
> phaseportrait(de1, [x], t=0..2, {{[0,1], [0,2]}});
```

This command will plot solutions of the differential equation for the two initial conditions $x(0) = 1$ and $x(0) = 2$ over the range $0 \leq t \leq 2$. The direction field is also plotted by default. The syntax is similar to that of `dfieldplot`, but the fourth argument, which provides initial conditions, is required. Each of the two lists in the fourth argument provides a value of $t$ and a value of $x$ to be used as initial conditions. The $t$ value must come first. The `arrows` option, as described previously, can be used to modify or remove the direction field.

### 2.4.2 Two-dimensional autonomous systems

Here is an example using `phaseportrait` to plot two trajectories for our two dimensional system, equation 5.

```latex
> phaseportrait(sys1, [x, y], t=0..1, {{[0,0,1], [0,1,0]}, x=-2..2, y=-2..2});
```

The first four arguments to `phaseportrait` are required. First comes the system of differential equations. The second argument is the list of dependent variables. The third argument specifies a range for $t$. In the example, we specified the range $t = 0$ to $t = 1$, but you might have to use a larger or smaller range. The fourth argument is a set of lists. Each list in the set represents an initial condition, giving the values of $t$, $x$ and $y$. For example the list $[0,0,1]$ represents the initial condition $x(0) = 0$ and $y(0) = 1$. The first item in the list is the value of $t$. More initial conditions could have been specified by including more lists in the set.

The direction field is shown by default. You can modify the direction field plot by specifying a value for the `arrows` option. Here is an example.

```latex
> phaseportrait(sys1, [x, y], t=0..1, {{[0,0,1], [0,1,0], [0,1/2,-1/2]}, x=-2..2, y=-2..2, arrows=NONE});
```

The `arrows` argument can only have the values given above for the `dfieldplot` command. The value `SMALL` is the default.

In the previous examples we have used a relatively short time interval. Many times you need to use a longer time interval. This can result in a very jagged plot, like from the following command.

```latex
> phaseportrait(sys1, [x, y], t=0..5, {{[0,0,1], [0,1,0]}, x=-2..2, y=-2..2});
```

The reason for the jagged appearance is that the numerical integration is being done with a stepsize that is too large. The fix is to set the step size with the `stepsize` command.
as in the following command.

\[
\text{phaseportrait}(\text{sys1}, [x,y], t=0..5, \{[0,0,1],[0,1,0]\}, \text{x=}-2..2, \text{y=}-2..2, \text{stepsize=0.1});
\]

If the plot is still jagged after you decrease the step size, try decreasing it again. However, the smaller the step size, the longer the integration takes, so be cautious in reducing the step size.

Finally, the phaseportrait command can be used to plot individual components versus t by using the scene optional argument. Here is an example of plotting \(x\) versus \(t\) for our two-dimensional autonomous system.

\[
\text{phaseportrait}(\text{sys1}, [x,y], t=0..1, \{[0,0,1],[0,1,0]\}, \text{scene=[t,x]});
\]

You might wonder why you would want to do this. One common reason to plot a single component is to determine the approximate period when a solution is periodic.

2.4.3 Two-dimensional non-autonomous systems

To use the phaseportrait command with our non-autonomous example, equation 6, we first define the system with the following command.

\[
\text{sys2} := \{\frac{\partial}{\partial t} y(t) = -y(t) - 4x(t) + \sin(t), \frac{\partial}{\partial t} x(t) = y(t)\};
\]

Then we can plot a trajectory in the two dimensional \(xy\) space with the command

\[
\text{phaseportrait}(\text{sys2}, [x,y], t=0..30, \{[0,1,1]\}, \text{stepsize=0.1});
\]

However, it is often useful to plot one of the solution components versus time by using the scene parameter. The example below plots \(x(t)\) versus \(t\).

\[
\text{phaseportrait}(\text{sys2}, [x,y], t=0..30, \{[0,1,1]\}, \text{scene=[t,x]}, \text{stepsize=0.1});
\]

2.5 Lower level routines: DEplot and DEplot3d

The DEplot command does the same thing as dfieldplot and phaseportrait, but is more general. In fact, dfieldplot and phaseportrait just call DEplot with appropriate arguments. The DEplot3d command plots solution trajectories in three dimensions and cannot plot direction fields, but is otherwise very similar to DEplot. For further information, look at the help pages. Optional arguments that we have described earlier, like arrows, stepsize, and scene can also be used with these commands.

The DEplot command has two forms. The first is used to plot the direction field and a set of trajectories and the second form plots only the direction field. However, it is also possible to plot solution trajectories without the direction field. We give examples below for each behavior for our model, equation (4).

To plot a set of trajectories as well as the direction field, use a command like the following.

\[
\text{DEplot}(\text{de1}, [x], t=0..2, \{[0,1/2],[0,-1/2],[0,2]\});
\]

You can also set the range for the dependent variable, but it isn’t required. Instead, Maple computes the trajectories and then sets an appropriate range for \(x\).
To plot only the direction field, you omit the initial conditions and set a range for the dependent variable, like the following.

```maple
> DEplot(de1, [x], t=0..2, x=-2..2);
```
To plot only trajectories without the direction field, use the optional argument `arrows=NONE`. The default value of the `arrows` parameter is `SMALL`, but you can set it to one of the other values given above for the `phaseportrait` command. Here is an example using the `arrows=NONE` option.

```maple
> DEplot(de1, [x], t=0..2, {{0,1/2}, [0,-1/2], [0,2]}, arrows=NONE);
```

To plot a set of trajectories as well as the direction field, use a command like the following.

```maple
> DEplot(sys1, [x, y], t=0..1, {{0,0,1,0}, [0,1,0]});
```

If you don’t like the ranges that Maple chooses for the dependent variables, you can specify them yourself.

To plot just the direction field, leave out the initial conditions and specify ranges for the two independent variables.

```maple
> DEplot(sys1, [x, y], t=0..1, x=-2..2, y=-2..2);
```

To omit the direction field and just plot the trajectories, use the optional argument `arrows=NONE`.

```maple
> DEplot(sys1, [x, y], t=0..1, {{0,0,1,0}}, arrows=NONE);
```

To plot trajectories of our two-dimensional non-autonomous system in the three dimensional `txy` space, use the `DEplot3d` command. Here is an example.

```maple
> DEplot3d(sys2, [x, y], t=0..30, {{0,1,1}}, stepsize=0.1);
```