APPENDIX B

Maple notes

This chapter contains some notes to help you get started using Maple. There are several styles for sending input to Maple, selected by the drop-down at the top of the worksheet. The default is "2D input", so the input to raise 2 to the 100$^{\text{th}}$ power appears as 2^{100}. Since it is not always clear how to input things like fractions, exponents, differential equations, etc. I have represented input expressions as "Maple input". This is a text-only input format, so it is clear which keystrokes are necessary to indicate things like exponents, fractions, etc. For example, the input to raise 2 to the 100$^{\text{th}}$ power, as Maple input, appears as `2^100`.

You should be able to copy-and-paste from the input lines in this chapter to a Maple worksheet, even if the input mode in the worksheet is "2D input".

Maple comes with an extensive help system, and these notes are not meant as a replacement. You can find help on a particular topic by searching for it: You can get a table of contents and a search box by opening "Maple help" on the Help menu. You can also enter `help(DEplot)`, for example, to get help on the DEplot procedure. You can get the same effect if you position your cursor on a term in the worksheet and then hit the "F2" key or look under the "Help" menu.

B.1. Basic operations

B.2. Linear algebra

B.3. Differential equations

Here is an annotated Maple session which calculates and plots various quantities related to the system of differential equations

$$\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} y \\ -4x - .1y^2 \end{bmatrix}.$$

Load the `DEtools` package, and look at the help page:

```
> with(DEtools):
```

```
> help(DEtools);
```

You can define the system as a *set* (using `{...}`) or as a *list* (using `[...]`) of differential equations. It will be convenient to have a set later:

```
> DSys := {diff(x(t),t) = y(t)-2*x(t), diff(y(t),t) = -x(t) +
  .1*y(t)^3};
```

$$DSys := \left\{ \frac{d}{dt}x(t) = y(t) - 2\,x(t),\, \frac{d}{dt}y(t) = -x(t) + 0.1\,(y(t))^3 \right\}$$

You can ask Maple for an analytic solution, but you won't get anything useful:

```
> dsolve(DSys);
```

$$\Bigg[\Bigg\{x(t) = \_a \text{ where } \Bigg[\Bigg\{ \left(\frac{d}{d\_a}\_b(\_a)\right)\_b(\_a) - \frac{3}{50}\,(\_b(\_a))^2\,\_a + 1/5\,\_b(\_a) + 4\,\_a - 1/10\,(\_b(\_a))^3 - \frac{3}{250}\,\_b(\_a)\,\_a^2 - \frac{1}{1250}\,\_a^3 = 0\Bigg\},$$
$$\left\{\_a = x(t),\, \_b(\_a) = \frac{d}{dt}x(t)\right\}, \left\{t = \int (\_b(\_a))^{-1}\,d\_a + \_C1,\, x(t) = \_a\right\}\Bigg]\Bigg\},$$
$$\left\{y(t) = \frac{d}{dt}x(t) + 1/5\,x(t)\right\}\Bigg]$$

Now we are going to draw some orbits in the phase portrait. We will need to specify initial conditions for the different solutions. This must be set up as a list or a set; we'll use a list, because we might want to color the curves differently and in a list the order is fixed. Each initial condition is specified as a list, containing initial values for the variables.

These initial values were determined by experimentation: I specified one or two, looked at the graph, decided to put in a couple more or make adjustments, etc.

```
> Ics := [[x(0)=0,y(0)=2],[x(0)=2,y(0)=0],[x(0)=2,y(0)=2],
  [x(0)=2,y(0)=3],[x(0)=-4,y(0)=0],[x(0)=4.3,y(0)=0],
  [x(0)=4.4,y(0)=0]];
```

$$Ics := [[x(0) = 0, y(0) = 2], [x(0) = 2, y(0) = 0], [x(0) = 2, y(0) = 2], [x(0) = 2, y(0) = 3], [x(0) = -4, y(0) = 0], [x(0) = 4.3, y(0) = 0], [x(0) = 4.4, y(0) = 0]]$$

Now we ask for a plot. The `DEplot` procedure has several forms (look at the help); the form we are going to use requires that first six arguments, in order, are:

(1) The dynamical system.
(2) The variables to plot; this is `[x(t),y(t)]`.
(3) The range for t, in the form `t=firstT .. lastT`.
(4) The range for x, as above.
(5) The range for y, as above.
(6) The initial conditions.

There are also many options that can be specified after the first six arguments. We'll use:

(1) `color`, the color of the arrows,
(2) `linecolor`, the color of the trajectories.
(3) `thickness`, the thickness of the trajectories.
(4) `numpoints`, the number of points to plot per curve. Use a higher number if the curves seem jagged. The default is 49.

Another useful option is `obsrange`, which indicates whether trajectories that leave the plotting area should be redrawn if they re-enter. This is not necessary for the current example. See help for `DEplot`, and for `plot`, for other options.

Here is the plot statement. The x, y and t ranges were determined by experimentation. I also found the equilibrium points and made sure that they were included.

```
> DEplot(DSys,[x(t),y(t)],t=-5..10,x=-5..5,y=-5..5,Ics,color=gray,
  linecolor=blue,thickness=1,numpoints=500);
```

From the graph it seemed that there was something interesting going on for orbits through approximately $(4.3, 0)$. To look more carefully at this I ran a different DEplot command, with three changes. First, I changed the initial conditions, since I only wanted to look at two solutions. I just put the list of initial conditions in the plot statement, rather than defining a new variable. Second, I wanted different colors for the two graphs, so the `color` option now specifies a list. (Search the help for `colornames` for a list of the colors that Maple understands.) Third, I used the `scene` option to specify a graph of y versus t.

```
> DEplot(DSys,[x(t),y(t)],t=-5..10,x=-5..5,y=-5..5,
  [[x(0)=4.4,y(0)=0],[x(0)=4.3,y(0)=0]],color=gray,
  linecolor=[red,blue],thickness=1,numpoints=400,scene=[t,y]);
```

From this it seems that, as $t \to \infty$, $y(t) \to 0$ on one of the solutions but $y \to -\infty$ on the other. It is hard to quantify this from the graph, but Maple will give us a numeric version of the solution. This uses the `dsolve` procedure as above, but with a `numeric` option. Since we are looking at only one solution curve it is necessary to specify a single initial condition, and this version of `dsolve` requires that the differential equations and the initial conditions be specified in a single set. We defined `DSys` as a set of differential equations, so we just need to add the initial conditions to this set, which we do using the `union` operator:

```
> orb1 := dsolve(DSys union {x(0)=4.4,y(0)=0}, numeric, [x(t),y(t)]);
```

$$orb1 := \text{proc}\,(x\_rkf45)\,...\,\text{endproc}$$

The result is a *procedure*, which we can use like a function to find points on the solution curve. For example (after some experimentation):

```
> orb1(5);
```

$$[t = 5., x(t) = -1.34498191430922986, y(t) = -3.06222234749248834]$$

```
> orb1(5.6);
```

$$[t = 5.6, x(t) = -1.97384367918946269, y(t) = -7.27741244337618465]$$

```
> orb1(5.69);
```

$$[t = 5.69, x(t) = -2.60991725995272894, y(t) = -27.6725723616629864]$$

```
> orb1(5.696);
```

$$[t = 5.696, x(t) = -2.83579696162862049, y(t) = -96.8877322592397405]$$

```
> orb1(5.7);
```

Error, (in orb1) cannot evaluate the solution further right of 5.6965326, probably a singularity

It looks like the solution through $(4.4, 0)$ blows up.

We can set up the solution through $(4.3, 0)$ similarly:

```
> orb2 := dsolve(DSys union {x(0)=4.3,y(0)=0}, numeric, [x(t),y(t)]);
```

Then, after some experimentation, we see that $x(10) \approx -0.0161$ and $x(11) \approx -0.0069$. If we want to determine t so that $x(t) = -.01$ then it appears that t is between 10 and 11. To find a better approximation to t we can try searching further, but it is simpler to prepare a table of values. Rather than writing a loop to do this, we use another version of `dsolve` that computes a table of values. As part of the input we need to specify a sequence of t values to use, in the format `output = array([list of t values])`. In the following we use Maple's `seq` function to fill in the t values. The `seq` function generates a sequence from a formula, and we'll generate the t values at 0.1 intervals between 10 and 11. I told Maple to round the answers and display 10 digits: To do this, click on the "Tools ¿ Options" menu item, and select the "Precision" tab.

```
> dsolve(DSys union {x(0)=4.3, y(0)=0}, numeric, [x(t),y(t)],
  output = array([seq(10+i/10, i = 0 .. 10)]));
```

t	$x(t)$	$y(t)$
10.0	−0.0161789040	−0.0187261898
10.1000000000	−0.0148697645	−0.0171746947
10.2000000000	−0.0136632824	−0.0157489123
10.3000000000	−0.0125517681	−0.0144389579
10.4000000000	−0.0115280660	−0.0132356876
10.5000000000	−0.0105855389	−0.0121306775
10.6000000000	−0.0097180050	−0.0111161205
10.7000000000	−0.0089197184	−0.0101847950
10.8000000000	−0.0081853621	−0.0093300578
10.9000000000	−0.0075100013	−0.0085457726
11.0	−0.0068890505	−0.0078262601

Apparently $x(t) = -.01$ is satisfied for t somewhere between 10.5 and 10.6. If you regenerate the table with steps of .01 between 10.5 and 10.6 then you'll find that t is between 10.56 and 10.57.