EXPLORATIONS

CHAPTER 8

Differential Equations

One of the features which distinguishes the TI-86 (and its predecessor the TI-85) from other TI graphing calculators is the powerful built-in differential equation capabilities of the machine. In fact, these capabilities are a principal reason why the TI-85 and now the TI-86 are the recommended calculators for both the core calculus and differential equations courses in the authors' department.

New differential equation features on the TI-86, not found on the TI-85, include built-in slope field and direction field graphing capabilities, TABLE access for solutions, use of lists to specify more than one set of initial conditions, and an EXPLORE feature that lets you draw solutions to a differential equation by selecting the initial point directly from the graph screen. In addition, an Euler approximation algorithm has been added so that now the user can choose between Euler and Runge-Kutta.

In this chapter we give examples that illustrate the use of the TI-86 in analyzing first order, second order, and autonomous systems of two first order differential equations.

§1 – Graphing a Slope Field

In this section we introduce the DifEq graphing mode by having the TI-86 draw a slope field for the differential equation

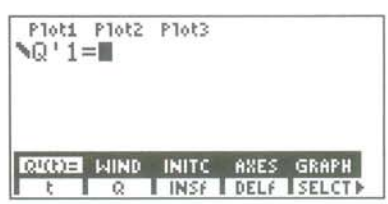
$$\frac{dQ}{dt} = t^2 + Q^2$$

in the viewing window $[-2, 2, 1] \times [-5, 5, 1]$.

1. First change the MODE setting to DifEq as shown in (8.1.1).

(8.1.1)

2. Then press [GRAPH] and select the ⟨**Q′(t)=**⟩ graph editor to obtain (8.1.2).



(8.1.2)

3. In DifEq mode, the generic variables are t and Q rather than the x and y in Func mode. In Func mode, we used $y1$, $y2$, etc. to name the functions. Similarly, in DifEq mode, we use $Q1$, $Q2$, etc. Consequently, we enter $t^2 + Q1^2$ for $Q'1$ as shown in (8.1.3).

(8.1.3)

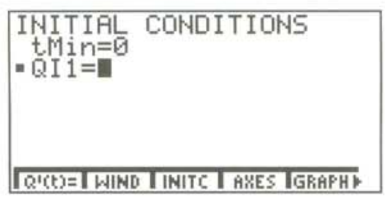
4. We are now ready to set up the TI-86 to graph a slope field for the differential equation in the [–2, 2, 1] × [–5, 5, 1] viewing window. With the display as in (8.1.3), access the graph FORMAT menu by pressing GRAPH MORE and selecting ⟨**FORMT**⟩. Make the necessary changes so that the FORMAT settings match those shown in (8.1.4). In particular, note that SlpFld is selected. (In DifEq mode we will frequently have to access the FORMAT menu to select the appropriate choice of SlpFld, DirFld, or FldOff.)

(8.1.4)

5. Next select ⟨**AXES**⟩ to obtain (8.1.5). This display indicates that the y-axis is $Q1$ (by default the x-axis is t), and that there will be 15 direction elements on each row of the slope field. The number for fldRes can be changed, but the default value of 15 is a good compromise.

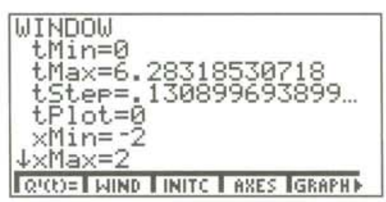
(8.1.5)

6. Select ⟨**INITC**⟩ to obtain a display similar to (8.1.6). The value you have for **tMin** does not have to be 0, but it is important that there be no value given for **QI1**. If your display has a value for **QI1**, press CLEAR to remove it. (If a value for **QI1** is given, then the TI-86 will attempt to graph the solution of the differential equation passing through the point (**tMin**, **QI1**) in addition to the slope field.)



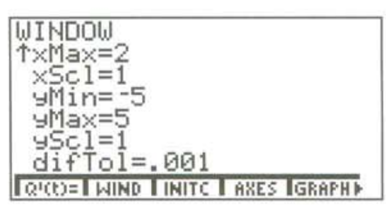
(8.1.6)

7. Then select ⟨**WIND**⟩ to obtain a display similar to (8.1.7). The values for **tMin**, **tMax**, **tStep**, and **tPlot** will have no impact on the graph of the slope field, so they do not have to match those of (8.1.7).



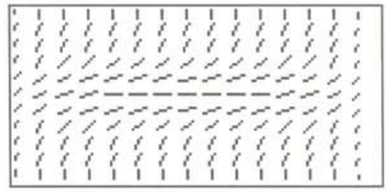
(8.1.7)

8. However, the WINDOW values for **xMin**, **xMax**, **xScl**, **yMin**, **yMax**, and **yScl** should match those shown in (8.1.7) and (8.1.8) since they establish the desired [–2, 2, 1] × [–5, 5, 1] viewing window. The value for **difTol** will not impact the slope field plot.



(8.1.8)

9. The setup to graph the desired slope field is complete, so select ⟨**GRAPH**⟩ to have the TI-86 graph the slope field shown in (8.1.9).



(8.1.9)

It is important to note that: *the last slope or direction field graphed is automatically stored by the TI-86 as a PIC picture named FldPic.* Later, in §6, we will use the FldPic picture to superimpose the slope field onto the graph of the solution of the differential equation in (8.1.3) satisfying the initial condition $Q1(-1) = 0$.

§2 – A First Order Initial Value Problem: Getting Started

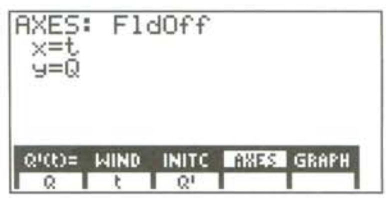
In the next few sections, we use the TI-86's built-in Runge-Kutta (RK) algorithm to calculate and graphically display in the viewing window $[-2, 2, 1] \times [-5, 5, 1]$ an approximate solution for the initial-value problem

$$\begin{cases} \dfrac{dQ1}{dt} = t^2 + Q1^2 \\ Q1(-1) = 0 \end{cases}.$$

1. First access the FORMAT menu and change the settings to agree with those shown in (8.2.1). In particular, note that AxesOn, RK, and FldOff have been selected.

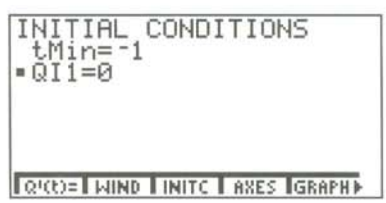
(8.2.1)

2. In §1 we entered the differential equation in the ⟨**Q′(t)=**⟩ graph editor, but the style indicated was the Thick style (the default style for DifEq mode). Change the style to Line style, and then select ⟨**AXES**⟩ to obtain what will probably be (8.2.2). This indicates that the horizontal axis will be the t-axis and the vertical axis will be the Q-axis. This is what we want, provided that $Q1$ is selected to be graphed. Alternatively, we can set $y = Q1$, and $Q1$ will be graphed even if it is not selected in the ⟨**Q′(t)=**⟩ graph editor. Now select ⟨**INITC**⟩ to obtain (8.1.6) again.



(8.2.2)

3. When graphing the solution of an initial-value problem, the value for **tMin** must be taken to be the initial t-value for the differential equation, and the value for **QI1** must be taken to be the initial $Q1$ value. Note that the **I** in **QI1** stands for *initial.* Also, the rectangular dot in front of **QI1** indicates we must assign **QI1** a value if we expect to graph a solution of an initial-value problem. Thus, assign **tMin** and **QI1** the values indicated in (8.2.3).

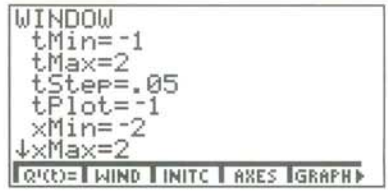


(8.2.3)

§3 – Graphing the Solution for $t > t_{initial}$

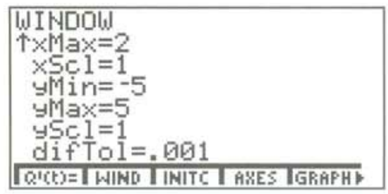
It is our goal to graph the solution of the initial-value problem introduced in §2 over the entire t-interval [–2, 2]. However, the nature of the algorithm only allows us to graph the solution on one side of **tMin** = –1 at a time. In this section, we will graph the solution over the t-interval [–1, 2]; and in §5, we will graph the solution over the t-interval [–2, –1]. It will then be possible to paste the two pieces together to obtain the graph over the entire t-interval [–2, 2]. See §6.

1. With the display as in (8.2.3), select 〈**WIND**〉 to obtain (8.1.7) again except this time the value of **tMin** will be –1. The largest value for t will be 2, so set **tMax** = 2. Set **tStep** = 0.05 and **tPlot** = –1. In most cases, **tPlot** is assigned the same value as **tMin**.



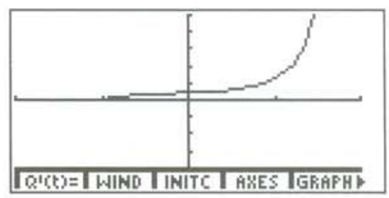
(8.3.1)

2. The values for **xMin**, **xMax**, **xScl**, **yMin**, **yMax**, and **yScl** set the viewing window for the solution. (These were set in §1.) Finally, set **difTol** = 0.001. This is the default value for **difTol**, and it will probably already be assigned this value. When completed, the WINDOW values should be as shown in (8.3.1) and (8.3.2). The setup is complete.



(8.3.2)

3. Select 〈**GRAPH**〉 to obtain (8.3.3), the graph of the solution over the t-interval [–1, 2].



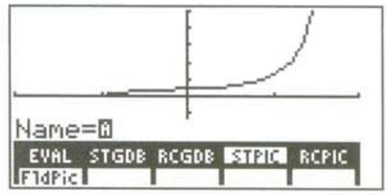
(8.3.3)

The **difTol** variable controls the accuracy of the RK approximate solution much as the **tol** variable controls the accuracy of certain computations (such as **fnInt**, **fMin**, and **fMax**). The **tStep** variable affects the resolution and appearance of the graph, but not the accuracy of the computed values when the RK method is used. In general, for the RK method, if the horizontal axis is the t-axis, setting **tStep** < (**xMax** – **xMin**)/126 increases plotting time without improving the resolution of the graph.

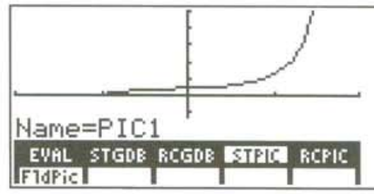
§4 –Solution Evaluations for $t > t_{initial}$

In this section, we store the graph displayed in (8.3.3) as PIC1 and approximate $Q1(t)$ for several values of t in the interval $[-1, 2]$.

1. With the display as in (8.3.3), press MORE MORE ⟨**STPIC**⟩ to obtain (8.4.1), and then type in PIC1 as shown in (8.4.2).

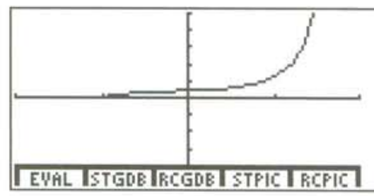


(8.4.1)



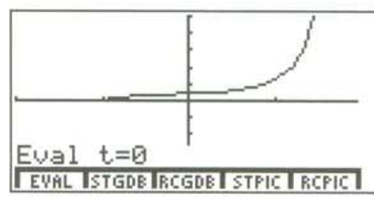
(8.4.2)

2. Then press ENTER to obtain (8.4.3) and to store the graph in PIC1.



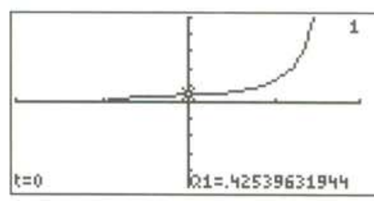
(8.4.3)

3. We can use the EVAL command to find approximate values of $Q1(t)$ for values of t in the interval $[-1, 2]$. *The accuracy of these approximations will be discussed in §8.* In (8.4.4) and (8.4.5) we show the displays that lead us to conclude that $Q1(0) \approx 0.42539631944$.



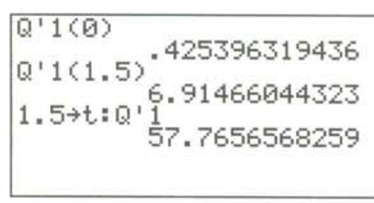
(8.4.4)

4. Similarly, we use EVAL to show that $Q1(1) \approx 1.2509959688$, $Q1(1.5) \approx 6.9146604432$, and that $Q1(1.7)$ is not defined. (It can be shown that the graph of $Q1(t)$ has a vertical asymptote near $t = 1.64$.)



(8.4.5)

5. Evaluation of $Q1(t)$ can also be done from the home screen but the notation is very confusing! To evaluate Q1(t) from the home screen, you must ask the TI-86 to calculate $Q'1(t)$. Display (8.4.6) shows some results from the home screen. This display also illustrates an inconsistency in that we do not get the same results when executing the commands $Q'1(1.5)$ and $1.5 \rightarrow t{:}Q'1$. (You can access the $Q'1$ symbol by selecting ⟨**EQU**⟩ from the CATALOG/VARIABLES menu.)

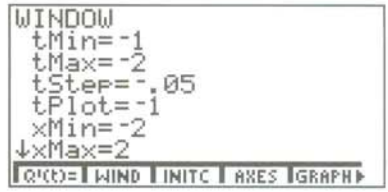


(8.4.6)

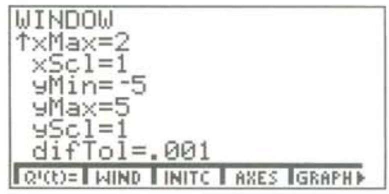
§5 – Graphing the Solution for $t < t_{initial}$

In order to graph the solution of the initial-value problem given in §2 over the t-interval $[-2, -1]$, we must interpret the roles of **tMin**, **tMax**, and **tStep** correctly. Actually the names **tMin**, and **tMax** are bad choices. Better names would have been **tInitial** and **tFinal**, respectively. With this observation you have probably guessed how we will graph the approximate solution over the t-interval $[-2, -1]$.

1. Return to the viewing window shown in (8.3.1) and (8.3.2). Change **tMax** to –2, **tStep** to –0.05, and leave the other settings the same. The displays (8.5.1) and (8.5.2) reflect these changes.

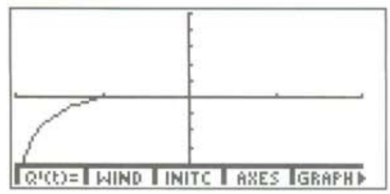


(8.5.1)



(8.5.2)

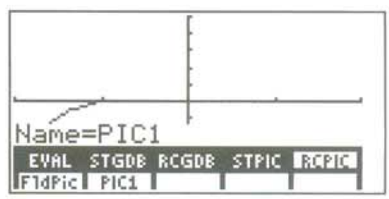
2. Then select ⟨**GRAPH**⟩. After a while you obtain (8.5.3). Then use EVAL as in §4 to conclude that Q1(–1.5) ≈ –0.8897869795, and Q1(–2) ≈ –6.686311581. (It can be shown that the graph of Q1(t) has a vertical asymptote near $t = -2.14$.)



(8.5.3)

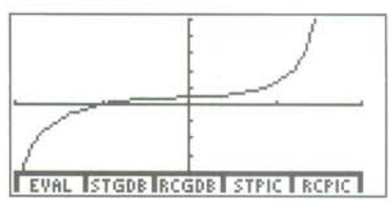
§6 – Superposition of the Two Pieces

1. Next we superimpose the graphs displayed in (8.3.3) and (8.5.3), and store the superposition as PIC1, intentionally overwriting what was previously stored as PIC1. With the display as in (8.5.3), press [MORE] [MORE] ⟨**RCPIC**⟩ ⟨**PIC1**⟩ to obtain (8.6.1).



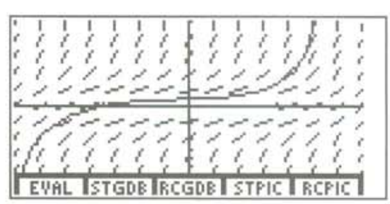
(8.6.1)

2. Then press [ENTER] to recall PIC1 and superimpose it on the graph given in (8.5.3). We obtain the display (8.6.2), which is a graph of the approximate solution over the t-interval $[-2, 2]$, *i.e.*, display (8.6.2) gives the graph of the approximate solution of the initial value problem given in §2 in the viewing window $[-2, 2, 1] \times [-5, 5, 1]$.



(8.6.2)

3. Store the graph displayed in (8.6.2) by pressing ⟨**STPIC**⟩ ⟨**PIC1**⟩ [ENTER]. This sequence of steps returns you to display (8.6.2). Finally, superimpose the slope field plot given in (8.1.9) with the display (8.6.2) by pressing ⟨**RCPIC**⟩ ⟨**FldPic**⟩ [ENTER] to obtain (8.6.3).



(8.6.3)

§7 – Comments on Euler's Method

If $Q(t)$ is the solution of the initial-value problem

$$\begin{cases} \dfrac{dQ}{dt} = F(t,Q) \\ Q(t_0) = Q_0 \end{cases}$$

and $t_n = t_0 + n\Delta t$, then Euler's method for approximating the solution $Q(t)$ consists of applying the iteration formula

$$Q_{n+1} = Q_n + (\Delta t)F(t_n, Q_n)$$

to compute successive approximations Q_1, Q_2, … to the true values $Q(t_1)$, $Q(t_2)$, … of Q.

The Euler method is not often used in practice because much more accurate methods (such as Runge-Kutta) are available. However, Euler's method is a popular starting point for discussing numerical solutions of differential equations due to its simplicity, and because the method yields insights into the workings of more accurate methods.

Setting up the TI-86 to approximate the solution of an initial-value problem using the Euler method is the same as when using Runge-Kutta except that the **difTol** setting in WINDOW is replaced by an **EStep** setting. The default **Estep** setting is 1, but it can be taken to be any integer from 1 to 25 inclusive. The Δt increment in the Euler method is given by

$$\Delta t = \frac{\textbf{tStep}}{\textbf{EStep}}.$$

Setting **Estep** > 1 can help the graphing speed since the TI-86 only plots points in increments of **tStep**. Unlike the Runge-Kutta method, the value of **tStep** (as well as **EStep**) affects the accuracy of the approximation.

If instead of Runge-Kutta, we use Euler with **tStep** = ±0.05, **EStep** = 5 and retrace what was done in §§ 2–6, then we will obtain a graph that is visually identical with that shown in (8.6.2). Moreover, using EVAL we obtain the Euler approximations:

$Q1(-2) \approx -6.027512949$, $Q1(-1.5) \approx -0.877492758$, $Q1(0) \approx 0.43177994321$,

$Q1(1) \approx 1.2560232242$, and $Q1(1.5) \approx 6.394302867$.

§8 – Euler and Runge-Kutta Comparisons

The table below indicates how accurate the Runge-Kutta and Euler methods are in approximating the solution $Q1(t)$ of the initial-value problem given in §2. In (1) the values given for $Q1(t)$ are correct to 7-decimal point accuracy. For this differential equation the Runge-Kutta method gives excellent approximations and the Euler method gives surprisingly good results. The Euler method starts to deteriorate as we use it to estimate $Q1(t)$ for t close to α or β, where $\alpha \approx -2.14$ and $\beta \approx 1.64$ are the t-coordinates for the two vertical asymptotes. Indeed, the Euler method will erroneously continue to give $Q1$-values for $t < \alpha$ and $t > \beta$.

	t	−2	−1.5	0	1	1.5
(1) (Exact)	**Q1**	−6.7037860	−0.8899551	0.4252398	1.2515292	6.9446615
(2) (difTol 0.001)	**Q1**	−6.6863116	−0.8897870	0.4253963	1.2509960	6.9146604
(3) (difTol 0.000001)	**Q1**	−6.7037629	−0.8899549	0.4252400	1.2515284	6.9446179
(4) (Euler $\Delta t = \pm 0.05$)	**Q1**	−4.5581676	−0.8313401	0.4580788	1.2775927	5.2016950
(5) (Euler $\Delta t = \pm 0.01$)	**Q1**	−6.0275129	−0.8774928	0.4317799	1.2560232	6.3943029
(6) (Euler $\Delta t = \pm 0.001$)	**Q1**	−6.6250270	−0.8886902	0.4258931	1.2519610	6.8808394

The speed of the RK method for comparable accuracy is vastly superior to the Euler method. Using the Euler method it takes about a minute to find $Q1(1.5) \approx 6.8808394$ when $\Delta t = 0.001$ and about ten minutes to find $Q1(1.5) \approx 6.9381736$ when $\Delta t = 0.0001$. On the other hand, using the RK method it takes less than eight seconds to find $Q1(1.5) \approx 6.9397788$ when **difTol** = 0.0001.

§9 – Graphing Solution Curves of a Predator-Prey System

In our next example, we use the TI-86 to investigate the solution of a system of two first order equations. Specifically, we study the initial-value problem

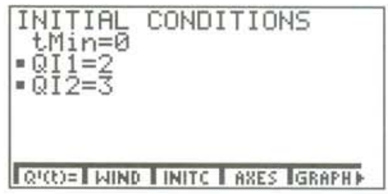
$$\begin{cases} Q'1 = Q1 - Q1*Q2 \\ Q'2 = -Q2 + Q1*Q2 \\ Q1(0) = 2, \quad Q2(0) = 3 \end{cases}.$$

This system is called a Lotka-Volterra system, and it is used to model a predator-prey environmental situation. To give meaning to the system at hand, we suppose that $Q1(t)$ is the number of millions of worms present t years from today and that $Q2(t)$ is the number of thousands of robins present t years from today. The initial conditions $Q1(0) = 2$, $Q2(0) = 3$ indicate that initially we have 2 million worms and 3 thousand robins. Explicit formulas for $Q1(t)$ and $Q2(t)$ cannot be obtained and so we must rely on numerical approximation techniques.

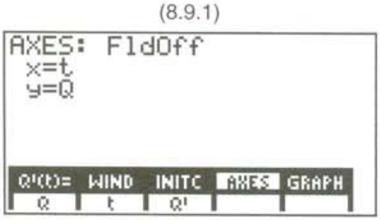
1. Set up the built-in algorithm as shown in (8.9.1)–(8.9.6).

(8.9.1)

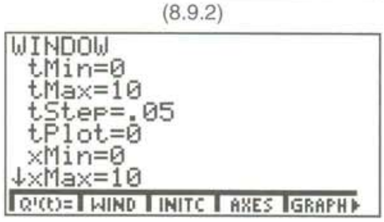
(8.9.2)



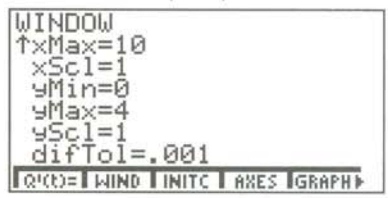
(8.9.3)



(8.9.4)

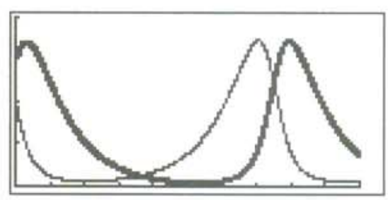


(8.9.5)



(8.9.6)

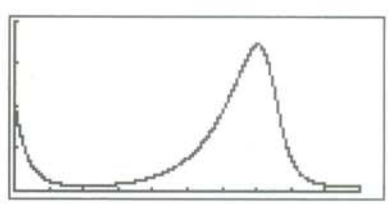
2. Select ⟨**GRAPH**⟩ to have the TI-86 graph the solutions $Q1(t)$ and $Q2(t)$ in the viewing window $[0, 10, 1] \times [0, 4, 1]$ as shown in (8.9.7). *Note that Q1 is graphed in Line style and that Q2 is graphed in Thick style.*



(8.9.7)

3. If we deselect the $Q'2$ equation for graphing purposes as shown in (8.9.8), but otherwise graph with the set-up given in (8.9.1)–(8.9.6), we obtain the graph of $Q1(t)$ in the viewing window $[0, 10, 1] \times [0, 4, 1]$ shown in (8.9.9). Similarly, we can deselect the $Q'1$ equation and select the $Q'2$ equation to obtain the graph of $Q2(t)$.

(8.9.8)

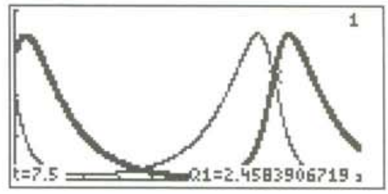


(8.9.9)

§10 – Periodicity of the Predator-Prey System

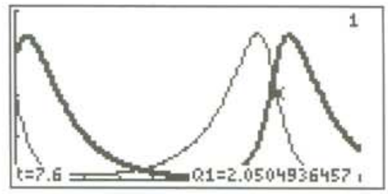
The general theory of Lotka-Volterra systems says that the solutions $Q1(t)$ and $Q2(t)$ are periodic and have the same period T. The display in (8.9.7) indicates that this period is about $T = 7.5$. We can use the TRACE feature of the built-in algorithm to obtain a better approximation for the period T.

1. Using the set-up given in (8.9.1)–(8.9.6) and the display as in (8.9.7), press GRAPH MORE ⟨**TRACE**⟩, type in 7.5, and press ENTER to obtain (8.10.1).



(8.10.1)

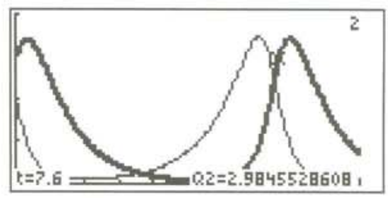
2. Then use the ▶ key to trace along the $Q1(t)$ curve until the $Q1$ value is as near 2 as possible. The resulting display is shown in (8.10.2).



(8.10.2)

3. Then press the ▲ key to obtain (8.10.3).

   The 1 in the upper right-hand corner of (8.10.1) indicates that we are tracing along the $Q1(t)$ curve. Similarly, the 2 in the upper right-hand corner of (8.10.3) indicates that we are tracing along the $Q2(t)$ curve.



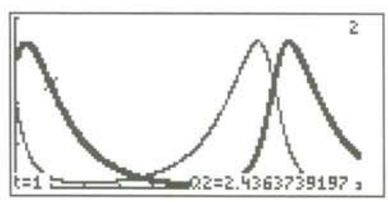
(8.10.3)

Since $Q1(0) = 2$, $Q2(0) = 3$, we deduce from (8.10.2) and (8.10.3) that $T \approx 7.6$. Additional analysis using a smaller **difTol** shows that $T \approx 7.603$ correct to three decimal places.

§11 – Further Analysis of the Predator-Prey System

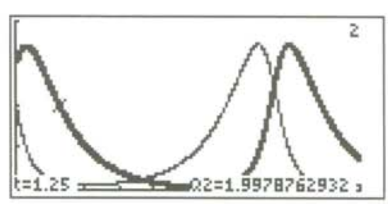
From (8.9.7), it appears possible that the graphs of $Q1(t)$ and $Q2(t)$ are the same, apart from a phase shift of about $t = 1$ unit. We can obtain a better approximation for this apparent phase shift by using TRACE as in §10.

1. With the display as in (8.10.3), type in 1 and press ENTER to obtain (8.11.1).



(8.11.1)

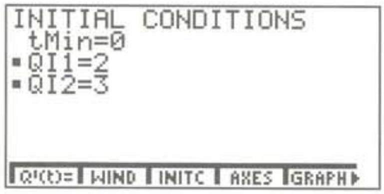
2. Then use the ▶ key to trace along the $Q2(t)$ curve until the $Q2$ value is as near 2 as possible. See (8.11.2). Since $Q1(0) = 2$, then it follows from (8.11.2) that the apparent phase shift is 1.25. However, as we now demonstrate, the $Q1(t)$ and $Q2(t)$ curves differ by more than a phase shift of 1.25 units.



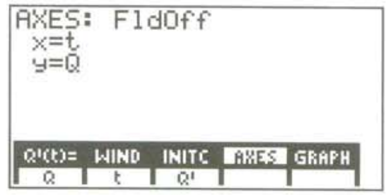
(8.11.2)

3. In (8.11.3)–(8.11.7), we set up the built-in algorithm to graph the $Q2(t)$ solution in the viewing window $[1.25, 11.25, 0] \times [0, 4, 1]$.

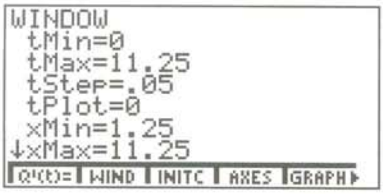
(8.11.3)



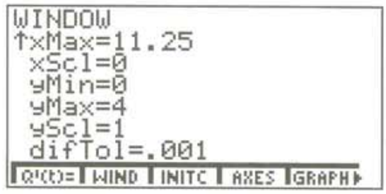
(8.11.4)



(8.11.5)

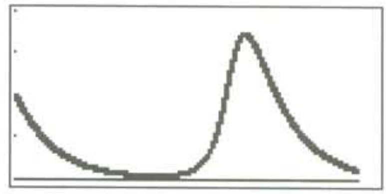


(8.11.6)



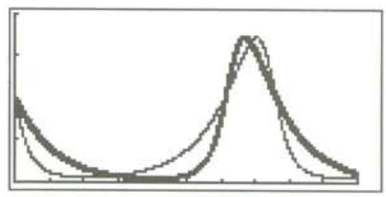
(8.11.7)

4. The resulting graph is displayed in (8.11.8).



(8.11.8)

5. Store the graph in (8.11.8) as a PIC file, repeat the part of §9 necessary to obtain (8.9.9) and then superimpose (8.11.8) on (8.9.9). You will obtain the display (8.11.9) which shows that $Q1(t)$ and $Q2(t)$ differ by more than a simple phase shift.



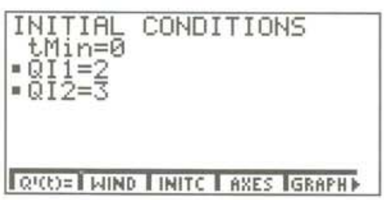
(8.11.9)

§12 – Graphing a Trajectory

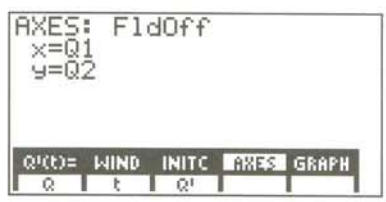
We can also use the built-in algorithm to graph $Q1$ versus $Q2$ for the Lotka-Volterra system in §9; *i.e.*, we can graph the trajectory for this system in the $Q1Q2$-phase plane.

1. Set up the algorithm as shown in (8.12.1)–(8.12.5).

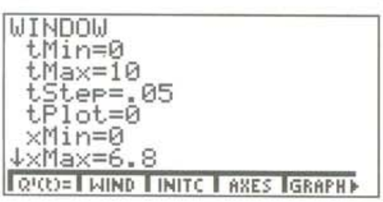
(8.12.1)



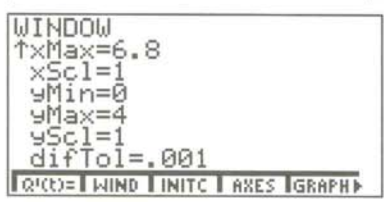
(8.12.2)



(8.12.3)

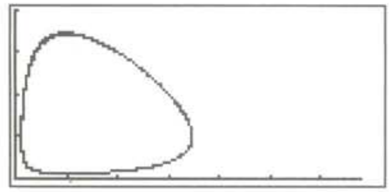


(8.12.4)



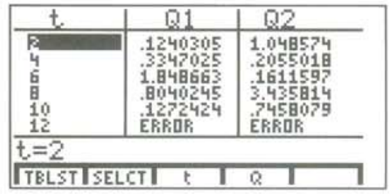
(8.12.5)

2. Then select ⟨**GRAPH**⟩ to have the TI-86 graph the trajectory in the viewing window $[0, 6.8, 1] \times [0, 4, 1]$ as shown in (8.12.6).



(8.12.6)

3. In (8.12.7), we see the table obtained by using TABLE with the settings in (8.12.1)–(8.12.5), **TblStart** = 2, and **ΔTbl** = 2. The ERROR results when $t = 12$ are due to the fact that **tMax** = 10 in our setup, and the TI-86 will calculate $Q1(t)$ and $Q2(t)$ only for t-values between **tMin** and **tMax**.



(8.12.7)

As you discovered waiting for the display (8.12.7) to appear, the TABLE feature is rather slow in DifEq graphing mode. For this reason, it is usually better to use EVAL or TRACE rather than TABLE when in DifEq graphing mode.

§13 – Graphing a Direction Field

An autonomous system of two first order equations is a system of the form

$$\begin{cases} \dfrac{dQ_1}{dt} = F(Q_1, Q_2) \\ \dfrac{dQ_2}{dt} = G(Q_1, Q_2) \end{cases}$$

in which the independent variable t does not explicitly appear in the functions F and G. For systems of this type, it is useful to construct a direction field. Such a direction field consists of a collection of line elements at points (Q_1, Q_2) in which the line element at point (Q_1, Q_2) has slope given by

$$\frac{dQ_2}{dQ_1} = \frac{G(Q_1, Q_2)}{F(Q_1, Q_2)}.$$

Thus, for autonomous systems we can use a direction field to help us visualize the trajectories for the system.

We illustrate by constructing a direction field in the viewing window $[0, 6.8, 1] \times [0, 4, 1]$ for the Lotka-Volterra system given in §9. As we will see, there are many similarities between setting up the calculator to do direction fields and slope fields. You may want to review §1 for comparison purposes.

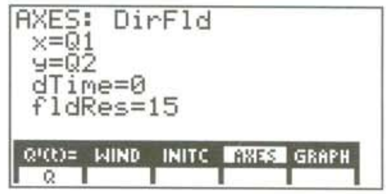
1. Begin by accessing the FORMAT menu and changing the settings to those given in (8.13.1) in which DirFld is selected.

(8.13.1)

2. Then select the 〈**Q′(t)=**〉 graph editor to obtain (8.13.2). Note that selecting DirFld in (8.13.1) evidently returned the styles associated with the two equations to the Thick style. (They were last set to Line style in (8.12.1).)

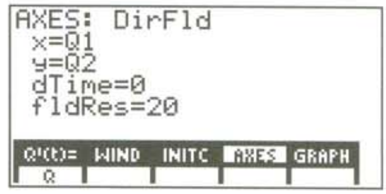
(8.13.2)

3. Next select 〈**AXES**〉 to obtain the default settings in (8.13.3). This display indicates that the y-axis is $Q2$ and that the x-axis is $Q1$. The value given for the dTime setting will have no effect on direction fields for autonomous systems. The effect this setting has for non-autonomous systems is beyond the scope of this chapter.



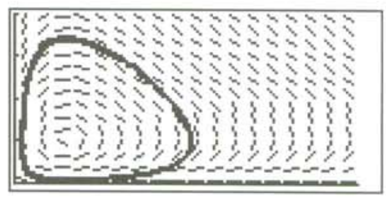
(8.13.3)

4. The fldRes setting has the same interpretation as it did for slope fields. Change fldRes to 20 as shown in (8.13.4).



(8.13.4)

5. Check that the other settings match those shown in (8.12.2), (8.12.4), and (8.12.5). Then select 〈**GRAPH**〉 to obtain the display (8.13.5). The TI-86 graphed both the direction field and the trajectory through the point (2,3).



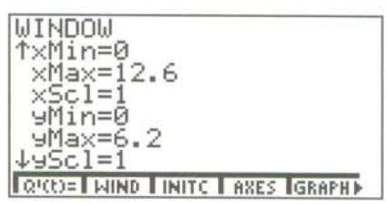
(8.13.5)

As indicated in §1, the graph of the direction field (but not the trajectory through the point (2, 3)) is stored by the TI-86 as a PIC picture named FldPic. If we only wanted to graph the direction field, then we would clear the values for **QI1** and **QI2** in (8.12.2). If we do this, then just as for slope fields, the values for **tMin**, **tMax**, **tStep**, and **tPlot** will have no impact on the graph of the direction field.

§14 – Using the EXPLORE Feature

The EXPLORE feature of the TI-86 allows us to interactively change the initial point by moving the cursor to a new initial point. Then we can direct the TI-86 to graph the solution passing through this new point. In this section we will use this feature to graph trajectories for the Lotka-Volterra system passing through points other than (2, 3).

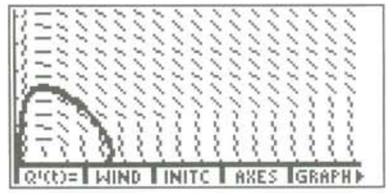
1. Since the EXPLORE feature requires that we move the cursor to the new initial point, it is convenient to select a viewing window in which the pixel lengths are "nice decimal lengths." Thus, we will change the viewing window [0, 6.8, 1] × [0, 4, 1], which gave us a viewing window with an aspect ratio of 1, to the viewing window [0, 12.6, 1] × [0, 6.2, 1], which will give us pixel lengths of 0.1 in both the horizontal and vertical directions. This change is shown in (8.14.1).



(8.14.1)

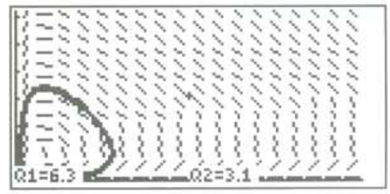
Differential Equations *(Continued)*

2. Leave all the other settings that resulted in display (8.13.5) unchanged. Then graph to obtain (8.14.2), which again gives us a direction field and the trajectory through the point (2, 3).



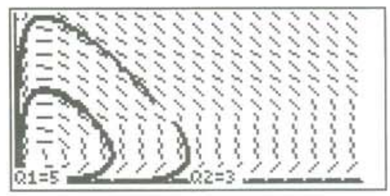
(8.14.2)

3. Then press MORE and select ⟨**EXPLR**⟩ to obtain (8.14.3).



(8.14.3)

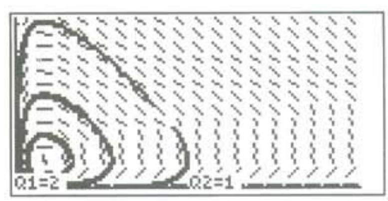
4. Use the arrow keys to move the free moving cursor to the point (5, 3), and press ENTER. The TI-86 draws the trajectory through the new point (5, 3) as shown in (8.14.4).



(8.14.4)

Note that the TI-86 graph of the trajectory through the point (5, 3) is not closed. This is because the EXPLORE feature uses the current settings of **tMin**, **tMax**, **tStep**, **tPlot** and **difTol** to draw the graph. In the case that **QI1** = 5 and **QI2** = 3, the period of the solutions $Q1(t)$ and $Q2(t)$ is greater than 10, so the trajectory is not closed. Selecting a large enough value for **tMax** would result in a closed trajectory.

5. Next, with the display as in (8.14.4), move the free moving cursor to the point (2, 1), and press ENTER to obtain the drawing of the trajectory through the point (2, 1) shown in (8.14.5).



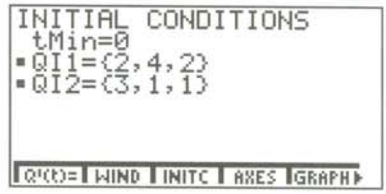
(8.14.5)

All graphs generated by the EXPLORE feature are *drawings* so we cannot use TRACE or EVAL on them. However, the EXPLORE feature is a nice way to quickly "explore" the solutions of a differential equation or a system of differential equations.

§15 – Lists of Initial Conditions

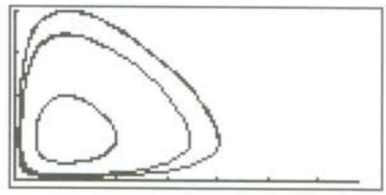
Until now we have used real numbers for the Q-values of the initial conditions. However, the built-in differential equations algorithm allows lists of real numbers to be used for these Q-values. To illustrate, we will use this feature to obtain for the Lotka-Volterra system the graph of the three trajectories through the points (2, 3), (4, 1), and (2, 1) in the viewing window $[0, 6.8, 1] \times [0, 4, 1]$.

1. Set up the algorithm the same as in (8.12.1)–(8.12.5), except use the lists {2, 4, 2} and {3, 1, 1} for **QI1** and **QI2**, respectively, as shown in (8.15.1). When the TI-86 sees the two lists for **QI1** and **QI2**, it knows to pair the ith entry of **QI1** with the ith entry of **QI2** to obtain the ith point through which a trajectory passes.



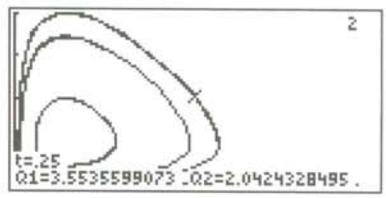
(8.15.1)

2. Thus, when ⟨**GRAPH**⟩ is selected, the TI-86 displays (8.15.2) by first graphing the trajectory through (2, 3), then the trajectory through (4, 1), and finally the trajectory through (2, 1).



(8.15.2)

3. Unlike the trajectories drawn using the EXPLORE feature, the trajectories here are graphs, and so we can use TRACE and EVAL on them. For example, in (8.15.3) we have used EVAL with $t = 0.25$ to show that the point (3.5535599073, 2.0424328495) is on the trajectory through the point (4, 1).



(8.15.3)

§16 – A Second Order Initial Value Problem

As our final example of this chapter, we use the built-in differential equations algorithm to calculate and graph the approximate solution of the second order initial-value problem

$$\begin{cases} \dfrac{d^2Q1}{dt^2} + t\dfrac{dQ1}{dt} + Q1 = 4\sin t \\ Q1(0) = 1, \;\; Q1'(0) = -1 \end{cases}$$

in the viewing window $[-10, 10, 1] \times [-3, 3, 1]$.

In order to use the built-in algorithm, we have to first convert the initial-value problem into an equivalent system of two first order equations.

This is done by introducing the additional function variable $Q2 = \dfrac{dQ1}{dt}$. The resulting equivalent first order system is then found to be

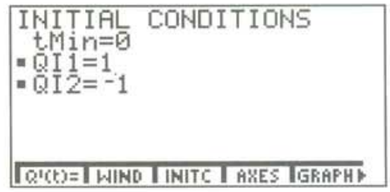
$$\begin{cases} \dfrac{dQ1}{dt} = Q2 \\ \dfrac{dQ2}{dt} = 4\sin t - Q1 - t*Q2. \\ Q1(0) = 1, \quad Q2(0) = -1 \end{cases}$$

§17 – Obtaining the Solution for $t > t_{initial}$

1. Set up the built-in algorithm as shown in (8.17.1)–(8.17.6).

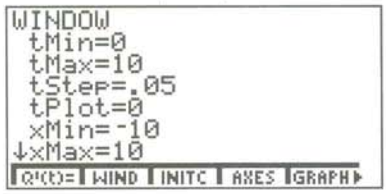
(8.17.1)

(8.17.2)

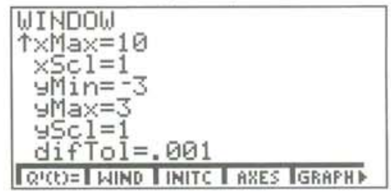


(8.17.3)

(8.17.4)

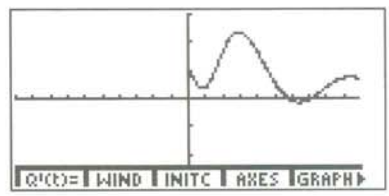


(8.17.5)



(8.17.6)

2. Select ⟨**GRAPH**⟩ to have the TI-86 graph the portion of the solution $Q1(t)$ for $t \in [0, 10]$ shown in (8.17.7).

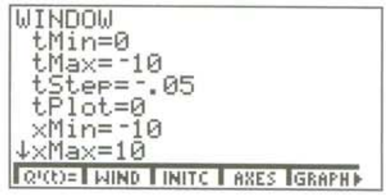


(8.17.7)

3. Then use EVAL to find that $Q1(3) \approx 2.346$, $Q1(6) \approx -0.090$, and $Q1(9) \approx 0.721$. If **difTol** is changed to 0.00001 and we regraph and use EVAL, we find that the values of $Q1(3)$, $Q1(6)$, and $Q1(9)$ differ from those when **difTol** = 0.001 by no more than one digit in the third decimal place. Thus, we are fairly confident that the given approximations of $Q1(t)$ are correct to nearly three decimal place accuracy. Finally, store the display (8.17.7) as PIC1.

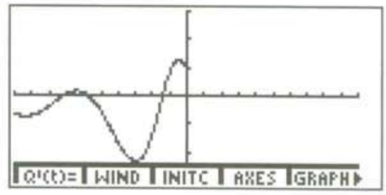
§18 – Obtaining the Solution for $t < t_{initial}$

1. Set up the built-in algorithm as shown in (8.17.1)–(8.17.6) except change (8.17.5) to (8.18.1).



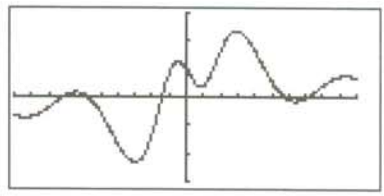
(8.18.1)

2. Select ⟨**GRAPH**⟩ to have the TI-86 graph the portion of the solution $Q1(t)$ for $t \in [-10, 0]$ shown in (8.18.2). Then use EVAL to find that $Q1(-3) \approx -2.324$, $Q2(-6) \approx 0.090$, and $Q3(-9) \approx -0.721$. As in §17, it appears that these evaluations are correct to almost three decimal place accuracy.



(8.18.2)

3. Finally, recall PIC1 and superimpose it on the display in (8.18.2) to obtain the graph of $Q1(t)$ in the viewing window $[-10, 10, 1] \times [-3, 3, 1]$ displayed in (8.18.3).



(8.18.3)

Exercises

1. Draw a slope field for each of the following differential equations in the indicated viewing window.

   (a) $\dfrac{dQ1}{dt} = 1 - 2t*Q1$ in the viewing window $[-1, 5, 1] \times [-1, 1, 1]$.

   (b) $\dfrac{dQ1}{dt} = -t^2 + \sin Q1$ in the viewing window $[-3, 3, 1] \times [-3, 3, 1]$.

2. Graph the solution of the initial-value problem

$$\begin{cases} \dfrac{dQ1}{dt} = 1 - 2t*Q1 \\ Q1(1/4) = 0 \end{cases}$$

   in the viewing window $[-1, 5, 1] \times [-1, 1, 1]$. Use Runge-Kutta with **difTol** = 0.001 and graph both sides of the solution. Also, use ⟨**EVAL**⟩ to find $Q1(-1/2)$, $Q1(1)$, and $Q1(4)$.

3. Graph the solution of the initial-value problem

$$\begin{cases} \dfrac{dQ1}{dt} = -t^2 + \sin Q1 \\ Q1(0) = 2 \end{cases}$$

   in the viewing window $[-3, 3, 1] \times [-3, 3, 1]$. Use Runge-Kutta with **difTol** = 0.001 and graph both sides of the solution. Use ⟨**EVAL**⟩ to find $Q1(-2)$, $Q1(2)$, and $Q1(3)$.

4. The system

$$\begin{cases} \dfrac{dQ1}{dt} = 5Q1 - Q1^2 - Q1*Q2 \\ \dfrac{dQ2}{dt} = -2Q2 + Q1*Q2 \end{cases}$$

has an asymptotically stable spiral point at $(Q1, Q2) = (2, 3)$.

(a) Graph the trajectory passing through the point $(Q1, Q2) = (6, 4)$ in the viewing window $[0, 8, 1] \times [0, 8, 1]$. Use Runge-Kutta with **difTol** = 0.001 and a t-interval of $[0, 10]$. Use ⟨**EVAL**⟩ to find $Q1(t)$ and $Q2(t)$ for $t = 1, 2, 5$, and 10.

(b) Graph the direction field in the viewing window $[0, 8, 1] \times [0, 8, 1]$ with **fldRes** = 20, and use ⟨**EXPLR**⟩ to draw several more trajectories for the system.

5. Airy's differential equation is given by

$$\frac{d^2Q1}{dt^2} - t*Q1 = 0.$$

Graph the solution of the initial-value problem consisting of Airy's equation and the initial conditions $Q1(0) = 0$, $Q1'(0) = 1$ in the viewing window $[-10, 3, 1] \times [-3, 3, 1]$. Use Runge-Kutta with **difTol** = 0.001, and graph both sides of the solution.

6. If $Q1(t)$ is the solution of the initial-value problem given in Exercise 5, then the indicated values of $Q1(t)$ given in (1) of the table below are correct to 5-decimal place accuracy. Use **difTol** = 0.001 and **difTol** = 0.00001 to verify the TI-86 results in (2) and (3) of the table. It is assumed that $[0, 3]$ is the t-interval when using ⟨**EVAL**⟩ to find $Q1(2)$ and $Q1(3)$; and that $[-10, 0]$ is the t-interval when using ⟨**EVAL**⟩ to find $Q1(-10)$, $Q1(-5)$, and $Q1(-2)$.

	t	−10	−5	−2	2	3
(1) (Exact)	**Q1**	−0.42872	−0.83195	−0.89918	3.61107	15.64385
(2) (**difTol** 0.001)	**Q1**	−0.41535	−0.82586	−0.89728	3.60764	15.61363
(3) (**difTol** 0.00001)	**Q1**	−0.42859	−0.83188	−0.89916	3.61103	15.64347

7. An important special function for engineers and scientists is the function $J_o(t)$, called the Bessel function of the first kind of order zero. If $\alpha > 0$, it can be shown that the solution $Q1(t)$ of the initial-value problem

$$\begin{cases} \dfrac{d^2Q1}{dt^2} + \left(\dfrac{1}{t}\right)\dfrac{dQ1}{dt} + Q1 = 0, \quad t > 0 \\ Q1(\alpha) = 1, \quad Q1'(\alpha) = 0 \end{cases}$$

converges to $J_o(t)$, as $\alpha \to 0$. Set α = 1E−999 (1E−999 is the smallest positive number that the TI-86 does not consider to be 0), and graph $Q1(t)$ in the viewing window $[0, 10, 1] \times [-2, 2, 1]$ using **difTol** = 0.00001. The values for $J_o(t)$ given in the table below are correct to 4-decimal place accuracy. Use ⟨**EVAL**⟩ to verify the values for $Q1(t)$.

	t	0.1	1	4	7	10
	$\mathbf{J_o}$	0.9975	0.7652	−0.3971	0.3001	−0.2459
(**difTol** 0.00001)	**Q1**	0.9975	0.7651	−0.3971	0.3000	−0.2458