Chapter 8
Elementary Differential Equations

The activities in this chapter present some additional things that we can now do with differential equations on this family of calculators. This complements the largely hand methods of Chapter 8 of the text *Calculus with Early Vectors*, by Phillip Zenor, Edward Slaminka, and Donald Thaxton, Prentice Hall, 1999.

1. Symbolic Differential Equations

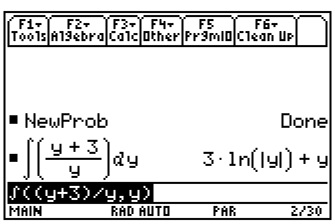
Section 1 of the text presents one method for solving some first-order differential equations, namely the "method of separation of variables." Assuming you can do the "separation" algebraically by hand, the task reduces to two indefinite integration problems. Obviously the symbolic integration on the TI-89 can be helpful.

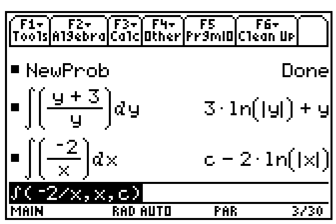
Consider $2y+(xy+3x)\dfrac{dy}{dx}=0$. Working by hand we find

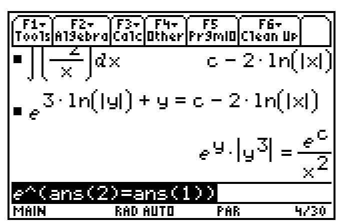
$$x(y+3)\frac{dy}{dx}=-2y$$

$$\frac{y+3}{y}dy=-\frac{2}{x}dx$$

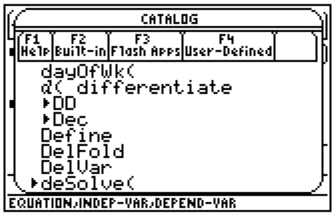
with the last line showing the separation of the two variables. Integrating both sides can be done by hand or by the calculator.

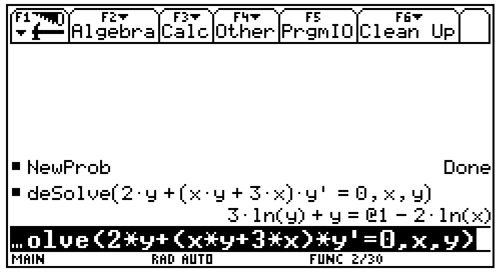


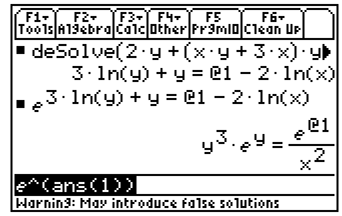




Letting $k=\pm e^{c}$ and dropping the absolute value, we would generally express the solution as $x^2y^3e^y=k$, which implicitly defines *y* as a function of *x*. The calculator can directly solve some differential equations, including the example above in a similar fashion. The command deSolve can be found in the F3 Calc menu or in the catalog.

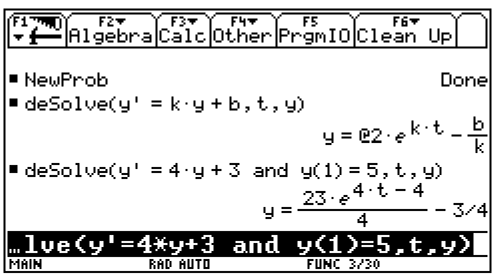


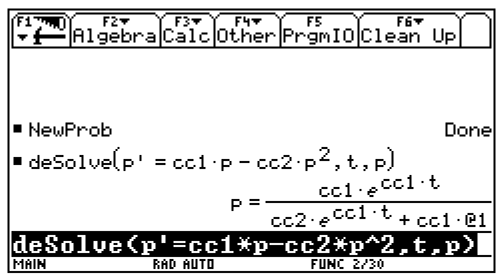




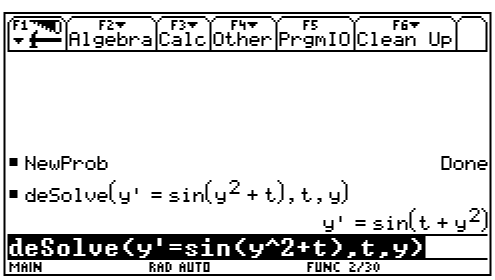
Notice the arbitrary constant in the general solution is represented by the variable "@1" in the output. Here the calculator is not as careful with absolute values as we were above.

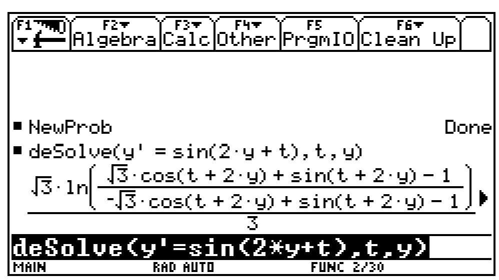
We can also ask the calculator to give us the symbolic solution for the general differential equation studied in Section 8.2, namely $y'(t) = k\ y(t) + b$. The command deSolve can be used to solve for a particular solution for such an equation with an initial condition. Notice below when we solve the general logistic differential equation, we cannot use the variable names "c1" and "c2" because they are reserved variables use by the system (for the columns in a data set or matrix). We use "cc1" and "cc2" instead for the constants in (12), page 408.





The command deSolve can solve some second-order differential equations as well. If it cannot solve a differential equation, it will just repeat the problem back as a result. Even when it can solve a problem with an implicit solution (such as $y' = \sin(2y + t)$ below), the solution may not be all that useful.

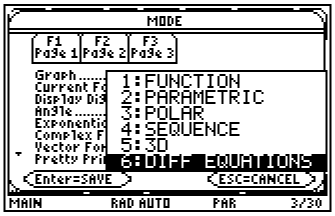


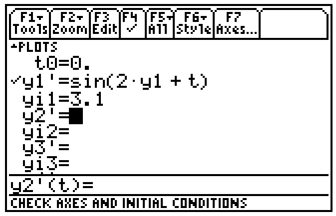


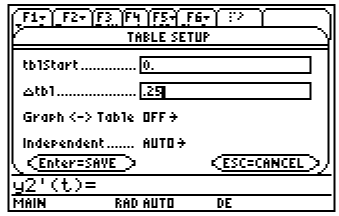
2. Differential Equation Graphing

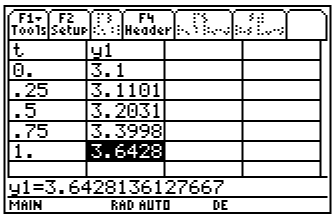
There is a differential equation graphing mode that allows the numerical solution for first-order differential equations with one or more initial conditions. We demonstrate below for a single equation first.

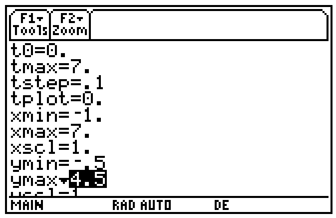
Consider $y' = \sin(2y + t), \quad y(0) = 3.1$

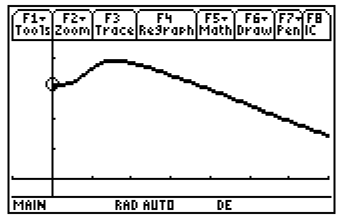






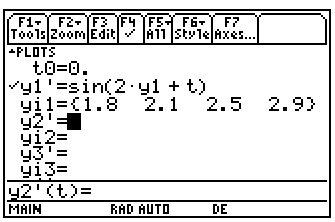


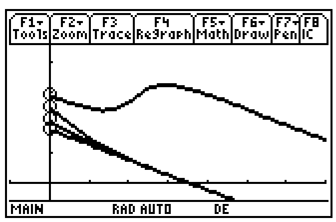


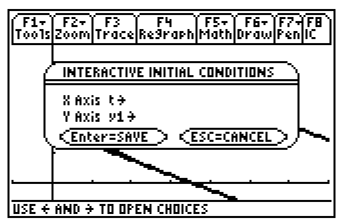


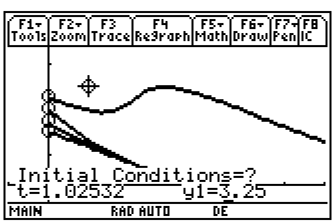
F1 9:Format Fields FLDOFF

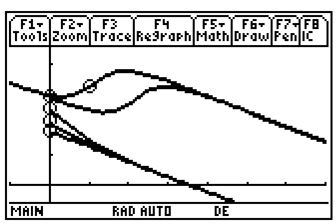
We can look at other solutions to this differential equations, either by entering in a list of initial values in the Y= edit screen or by pressing F8 IC to select further initial conditions interactively in the graph screen.











In a later differential equation course, you will study higher order differential equations and systems of first-order equations where the other features of this family of calculators will be used nicely.

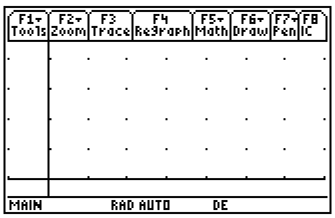
3. Slope Fields and Euler's Method

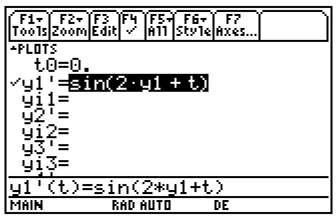
Many calculus texts now present the graphical understanding of first-order differential equations by considering a slope field for the general solution in some viewing window. The text actually has a slope field plotted in Figure 1.c, page 409 (where it is called a direction field). We present the general idea of a slope field below and introduce the simplest numerical method for solving a first-order differential equation with an initial condition, called Euler's method, which is closely related to a slope field.

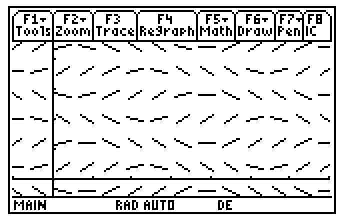
Consider a general first-order differential equation of the form

$$\frac{dy}{dt}(t) = f\left(t, y(t)\right).$$

Suppose the graph of the solution passes through the point (a, b), i.e. when $t = a$ then $y(a) = b$. We know from the differential equation that the solution must have $y'(a) = f(a,b)$. Given a desired viewing window, we construct a grid of points in the rectangle. We treat the coordinates of each point in our grid as the pair (a, b), and we draw a short line segment through the grid point with the slope $y'(a) = f(a,b)$ that the solution must have (if it goes through that grid point).

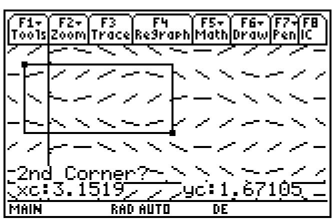


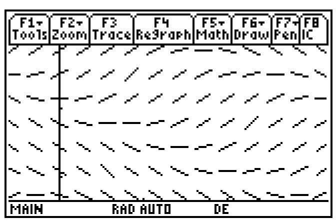


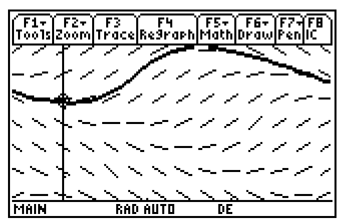


Sample Grid (Grid ON) DE with no initial condition F1 9:Format Field SLPFLD
(default grid, not as before)

For example, one of the grid points used in the last image seems to be about (0.87, 2.79). Thus that line segment is drawn with slope approximately sin(2*2.79 + 0.87) ≈ 0.166

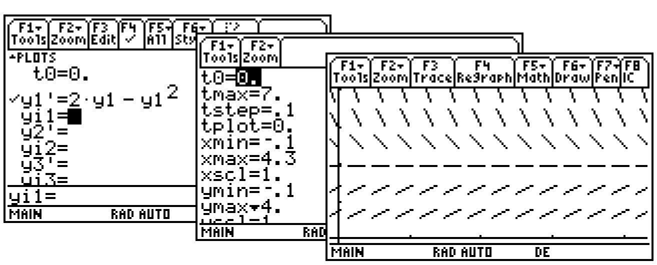


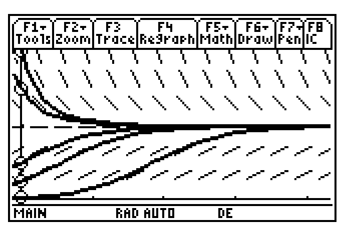




F2 Zoom ZoomBox More detail Add initial condition $y(0) = 3.1$

We reproduce the slope filed in Figure 1.c, page 409 for $P'(t) = 2P(t) - \left(P(t)\right)^2$.





$y'(0) = \{0.05, 0.5, 1.0, 3.0, 4.0\}$

TI uses an adaptive Runge-Kutta routine to numerically approximate the solutions in these differential equation plots by default. This routine works to try to get plotted points with an accuracy of diftol, a parameter you can set in the Window screen. The parameter fldres in the Window screen sets the grid spacing for the slope filed. The defaults which often work nicely are diftol = 0.001 and fldres = 14 (giving a 14 by 7 array of slope lines in the slope field).

The Graph Format screen allows you to select this adaptive method (RK) and a much simplier (less accurate) method (EULER). Euler's method is now taught in many calculus texts, and it is a required topic in the Advanced Placement Calculus course commonly taught in high schools. This simple method is based upon the fact that knowing the slope enables us to compute a tangent line approximation to the solution, which will be fairly accurate if we do not move the *t*-value too far from the known point.

To implement Euler's method for $\frac{dy}{dt}(t) = f\left(t, y(t)\right), \quad y(t_0) = y_0$, with step size *h*, we first compute the slope $m_0 = f(t_0, y_0)$. The line tangent to the graph of the solution we want at the point (t_0, y_0) will have the equation $y = y_0 + m_0\left(t - t_0\right)$. Thus for a small step *h*, the value predicted on the tangent line, namely $y = y_0 + m_0\left((t_0 + h) - t_0\right) = y_0 + m_0 h$, will be close to the solution $y(t_0 + h)$. We have started with $(a,b) = (t_0, y_0)$, used the slope and a short segment of the tangent line, and arrived at the point $(t_0 + h, y_0 + m_0 h)$ which we label (t_1, y_1). Then we treat (t_1, y_1) as (a,b), compute a new slope $m_1 = f(t_1, y_1)$, and move another step *h* along the new tangent line to estimate $y(t_2) = y(t_0 + 2h) \approx y_1 + m_1 h$. Continuing in this fashion, we recursively define sequences with

$$t_0 \text{ and } y_0 \text{ given}$$

$$t_{i+1} = t_i + h, \quad y_{i+1} = y_i + f(t_i, y_i)h$$

where $y(t_i) \approx y_i$.

For example, with $y' = \sin\left(2y + t\right), \quad y(0) = 3.1$ and *h* = 0.1, we get

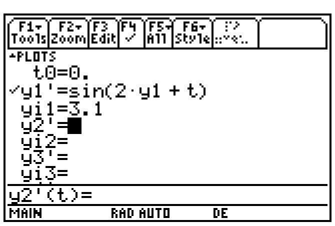
$$t_0 = 0, \quad y_0 = 3.1$$

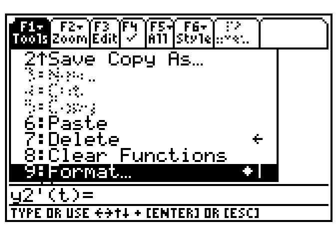
$$t_1 = 0.1, \quad y_1 = 3.1 + \sin(2*3.1 + 0)*0.1 \approx 3.09169$$

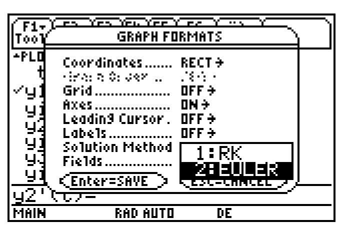
$$t_2 = 0.2, \quad y_1 = 3.09169 + \sin(2*3.09169 + 0.1)*0.1 \approx 3.09171$$

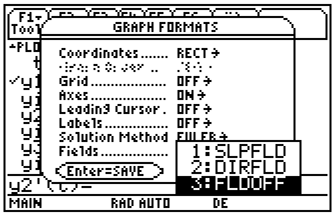
$$t_3 = 0.3, \quad y_1 = 3.09171 + \sin(2*3.09171 + 0.2)*0.1 \approx 3.10172$$

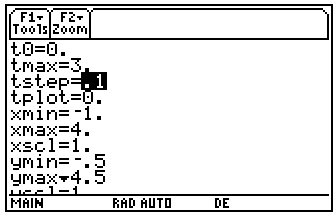
It is possible to have the calculator do these computations, display them in a table, and plot them as an approximate solution.

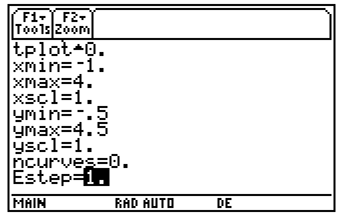




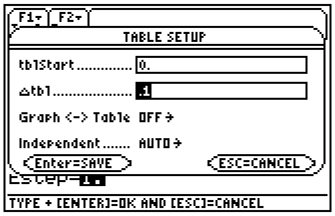


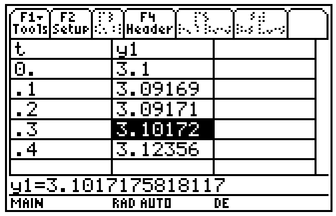


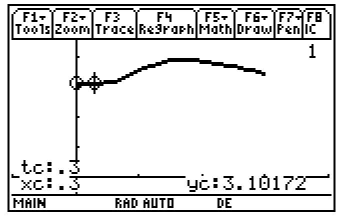




With Estep = 1, tstep = *h*







In general, Euler's method needs a very small step size *h* to be accurate. Thus you might not want to plot all of the values that you are computing. Choosing Estep = 5 and tstep = 0.1 will effectively use *h* = 0.1/5 = 0.02, even though the graph still only plots every 0.1. Thus Estep = 5 causes 5 Euler steps between each plotted point. Since you will be using Euler's method only when someone asks you to do so, I would expect you to always use Estep = 1 with the plotted points the desired answer (which can also be displayed in a table). If the method is not specified by your assignment, the select the RK method instead, picking diftol to be the desired tolerance.

What the textbook called "Euler's method" in Chapter 7 (page 338) is really just a special case of the more general method we have presented here when it happens that the differential equation does not explicitly depend upon *y*. Thus solving $F(t) = \int_c^t f(u)du$ for $t = c, c+h, c+2h, \ldots$ is equivalent to solving $y'(t) = f(t), \quad y(c) = 0$ using $t_0 = c, t_1 = c+h, t_2 = c+2h, \ldots$. This special case is also equivalent to looking at partial sums of the Riemann sum approximation for the definite integral with all of the evaluation points selected as left endpoints on an equally spaced partition.