EXPLORATIONS

APPENDIX D

Pathological Examples

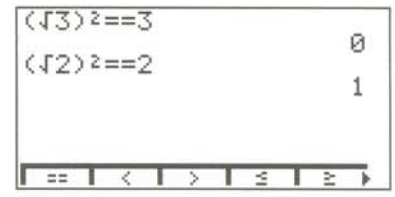
Due primarily to architectural and numerical algorithmic limitations, there are examples for which graphing calculators sometimes give surprising, misleading, or incorrect results. We refer to such examples as pathological examples. Several such examples have been included in this book. (See §13 and §16 of Chapter 2, §11 and §14 of Chapter 3, §9 of Chapter 5, and §1, §10 and §11 of Chapter 6.) In this appendix, we will investigate a few more of the interesting pathological examples we have encountered while using the TI-86.

§1 – Introduction

Any numerical algorithm can be fooled by sufficiently exotic examples, but the examples discussed here do not fall into that category. These examples are ones that we have routinely encountered in everyday use of the TI-86 and its predecessor the TI-85 as a teaching tool in our classes. We also want to mention that these pathological examples do not in any way diminish our enthusiasm for the TI-86. As will be seen later in this appendix, even a sophisticated software package, such as *Mathematica*, has difficulties similar to those of the TI-86 with some of these examples. Given its minimal cost and its portability, we consider the TI-86 to be an excellent computing device and a very valuable tool to be used in the undergraduate mathematics curriculum. It is also well suited to assist itself in analyzing what has happened with these problems. The examples included here can be thought of as starting points for interesting and valuable mathematical discussions in a variety of undergraduate courses.

The TI-86, just as any other calculator or computer, does finite arithmetic. In particular, the TI-86 generally keeps track of 14 digits and displays 12 digits. The two digits which are not displayed are often called guard digits. It is important to observe that the TI-86 cannot represent exactly any irrational real number since any such number will not have a decimal representation which terminates. Additionally, round-off error is always possible in doing numerical computations, and it can create unusual results.

For example, (D.1.1) shows that the TI-86 does not think $\left(\sqrt{3}\right)^2$ is equal to 3, but it does think $\left(\sqrt{2}\right)^2$ is equal to 2.



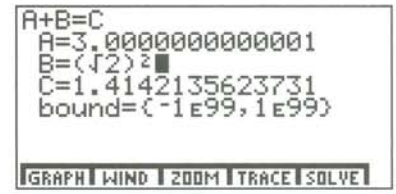
(D.1.1)

Round-off error is a factor here, but the finite arithmetic is actually more prominent in this case. The TI-86 considers two numbers to be equal if the internal 14-digit representations of the two numbers

are identical. Consequently, the calculator thinks $\left(\sqrt{3}\right)^2$ is not equal to 3 because the square of the calculator's value for $\sqrt{3}$ is not three followed by a decimal point and 13 zeros. In fact, there is no 14-digit number whose square rounded to 14 digits is 3 followed by a decimal point and 13 zeros. On the other hand, the TI-86 value for the square root of 2 is 1.4142135623731, and the exact square of this number is 2.000 000 000 000 014 004 103 603 61 (spaces provided for ease of reading). So, when the TI-86 considers the 14-digit version of this number, it gets exactly 2.

The built in SOLVER on the TI-86 can often be used to see the full 14-digit versions of quantities. Figure (D.1.2) indicates one way this can be done. In the SOLVER, we created an equation $A + B = C$ and pressed [ENTER]. The particular equation is not important. It just gives us a way to study the TI-86's 14-digit values for quantities.

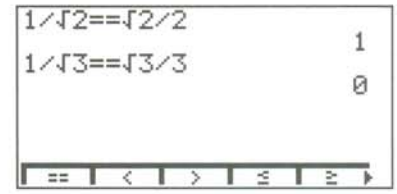
In fact, (D.1.2) shows the value the TI-86 has for $\left(\sqrt{3}\right)^2$ on the $A=$ line. The value for C was obtained by entering $\sqrt{2}$ followed by [ENTER]. The $B =$ line shows $\left(\sqrt{2}\right)^2$ for B just before pressing [ENTER].



(D.1.2)

Figures (D.1.3) and (D.1.4) show home screen images of two other unusual examples.

1. In (D.1.3), the TI-86 again seems to contradict standard rules of elementary algebra.



(D.1.3)

2. Figures (D.1.4) and (D.1.5) show the setup and the result of trying to compute 0^0 on the home screen. The TI-86 does not know a value for 0^0.

(D.1.4)

(D.1.5)

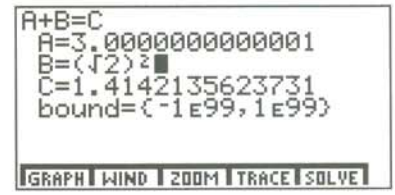
3. Figure (D.1.6) indicates that computing quantities like 0^3 and 5^0 are no problem.

(D.1.6)

are identical. Consequently, the calculator thinks $\left(\sqrt{3}\right)^2$ is not equal to 3 because the square of the calculator's value for $\sqrt{3}$ is not three followed by a decimal point and 13 zeros. In fact, there is no 14-digit number whose square rounded to 14 digits is 3 followed by a decimal point and 13 zeros. On the other hand, the TI-86 value for the square root of 2 is 1.4142135623731, and the exact square of this number is 2.000 000 000 000 014 004 103 603 61 (spaces provided for ease of reading). So, when the TI-86 considers the 14-digit version of this number, it gets exactly 2.

The built in SOLVER on the TI-86 can often be used to see the full 14-digit versions of quantities. Figure (D.1.2) indicates one way this can be done. In the SOLVER, we created an equation $A + B = C$ and pressed [ENTER]. The particular equation is not important. It just gives us a way to study the TI-86's 14-digit values for quantities.

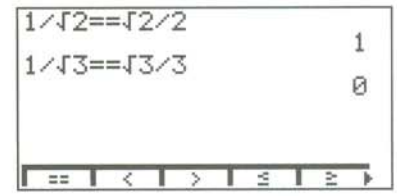
In fact, (D.1.2) shows the value the TI-86 has for $\left(\sqrt{3}\right)^2$ on the $A=$ line. The value for C was obtained by entering $\sqrt{2}$ followed by [ENTER]. The B = line shows $\left(\sqrt{2}\right)^2$ for B just before pressing [ENTER].



(D.1.2)

Figures (D.1.3) and (D.1.4) show home screen images of two other unusual examples.

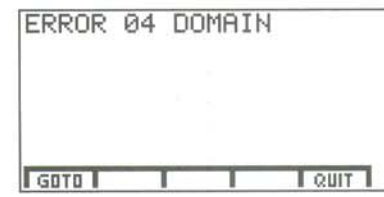
1. In (D.1.3), the TI-86 again seems to contradict standard rules of elementary algebra.



(D.1.3)

2. Figures (D.1.4) and (D.1.5) show the setup and the result of trying to compute 0^0 on the home screen. The TI-86 does not know a value for 0^0.

(D.1.4)



(D.1.5)

3. Figure (D.1.6) indicates that computing quantities like 0^3 and 5^0 are no problem.

(D.1.6)

§2 – Numerical Examples

The 14-digit accuracy capacity of the TI-86 can give rise to surprising results when working with the **int** and **mod** commands. For example, (D.2.1) shows the failure of the calculator to correctly report that

$$\mathbf{int}\left(\frac{10^{15}-1}{10^{12}}\right)=999.$$

The incorrect result occurs because the calculator thinks that $(10^{15}-1)$ is 10^{15}, as can be verified using the TEST menu similar to (D.1.1). The TI-86 computes $\mathbf{mod}(x, y)$ by using the formula $\mathbf{mod}(x, y)=x-y\,\mathbf{int}(x/y)$. Thus, as (D.2.1) shows, the TI-86 fails to correctly report that $\mathbf{mod}(10^{15}-1, 10^{12})=10^{12}-1$ since, as indicated above, it thinks that $(10^{15}-1)$ is 10^{15}.

Round-off error can also lead to other unusual results which seem to violate standard rules of algebra.

For example, (D.2.2) shows that $(e\text{^}1)^2$ is not equal to $e\text{^}2$ on the TI-86. Of course, these two expressions are computed differently by the calculator, the first being the square of the calculators value for $e\text{^}1$ and the second being an evaluation of the exponential function using the argument 2.

```
int ((10^15-1)/10^12)
                    1000
mod(10^15-1,10^12)
                       0
 NUM  PROB  ANGLE  HYP  MISC
 sign  min  max  mod
```

(D.2.1)

```
(e^1)²==e^2
                       0
 ==  <  >  ≤  ≥
```

(D.2.2)

The TI-86's ability to do complex arithmetic by default mode is a really nice and important feature for advanced mathematics classes, but it presents pitfalls for precalculus and beginning calculus courses. One of the first precalculus casualties is that the TI-86 often computes a value for a function at an argument that is supposedly not in the domain of the function.

For example, the TI-86 readily finds values for $\mathrm{Sin}^{-1}(2)$, $\ln(-1)$, and $f(-1)$ where $f(x)=\sqrt{x}$ as shown in (D.2.3). This apparent contradiction gives an opening to the idea that when the real number system is extended to the larger complex number system the domains of many of the standard functions in precalculus and calculus are likewise extended.

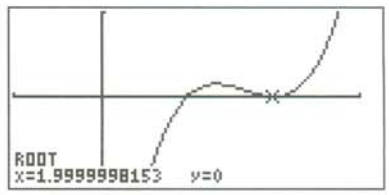
```
sin⁻¹ 2
(1.57079632679, -1.31...
ln (-1)
      (0,3.14159265359)
√(-1)
                  (0,1)
```

(D.2.3)

§3 – Graphing Examples

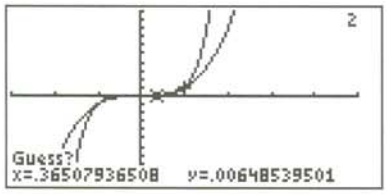
In §13 of Chapter 2 we considered an example where ⟨**ROOT**⟩ failed to find a multiple root at $x = 0$. To see that it is not just roots at $x = 0$ which cause problems, graph $y1 = (x-1)(x-2)^2$ in the viewing window $[-1, 3, 1] \times [-1, 1, 1]$ and try to use ⟨**ROOT**⟩ to find the root at $x = 2$. The result will be **ERROR 27 NO SIGN CHNG** unless you enter exactly 2 in response to **Guess**. Remarkably enough, however, round-off error can be of some assistance on this problem.

To see how, let $y2 = (y1 + 2) - 2$, and repeat the process of trying to find the root at $x = 2$ using **Left Bound** = –1.7, **Right Bound** = 2.1, and **Guess** = 1.75. The result of this computation will be similar to that shown in (D.3.1). Apparently there has been just enough round-off error to make the **ROOT** algorithm think that the necessary sign change has occurred.

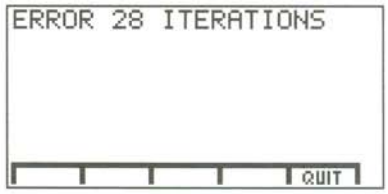


(D.3.1)

Since the ⟨**ISECT**⟩ algorithm utilizes many of the same solver capabilities of the TI-86 as the ⟨**ROOT**⟩ algorithm, it is not surprising that ⟨**ISECT**⟩ can have difficulties similar to those of ⟨**ROOT**⟩. For example, let $y1 = x\text{^}3$, and let $y2 = x\text{^}5$ in the viewing window $[-3, 5, 1] \times [-10, 10, 1]$. Of course, $x = 0$ is a multiple root of the equation $x^3 = x^5$. Figures (D.3.2) and (D.3.3) show the setup and the result of using the ⟨**ISECT**⟩ algorithm to try and find the point of intersection at (0,0).



(D.3.2)



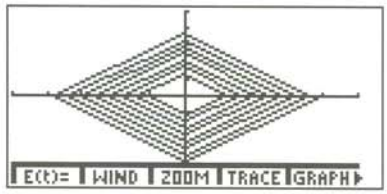
(D.3.3)

As mentioned in the beginning of Chapter 3, the ability of the TI-86 and other graphing calculators to easily graph parametric equations has made the study of parametric equations more prominent and more easily accessible for undergraduate mathematics courses. But as was seen in §11 of Chapter 3, there can be unexpected results which lead to important mathematical considerations. Consider again the following pair of equations

$$\begin{cases} x = t\cos(25\pi t) \\ y = t\sin(25\pi t) \end{cases}.$$

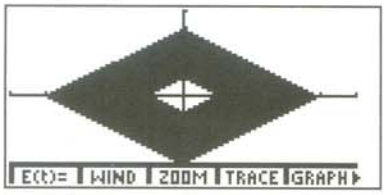
Figures (D.3.4) through (D.3.8) give the resulting graphs of this pair in the window $[-5, 5, 1] \times [-5, 5, 1]$ with $1 \le t \le 4$ for **tStep** being 0.05, 0.1, 0.01, 0.02, and $1/(25\pi)$, respectively. The wide variation in the graphs leads to interesting discussions as to what is the correct graph. It is a worthwhile exercise for beginning calculus students to show mathematically that the graph of this pair of equations is a tightly wound spiral.

(D.3.4)

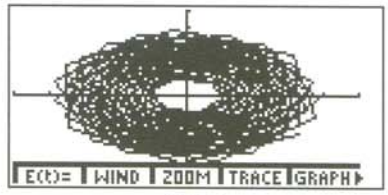


(D.3.5)

(D.3.6)



(D.3.7)

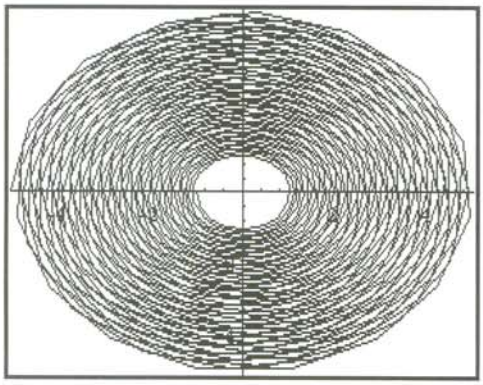


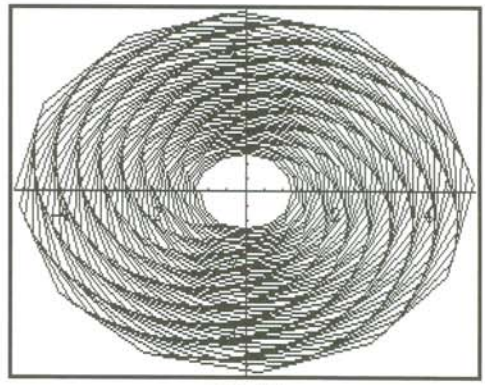
(D.3.8)

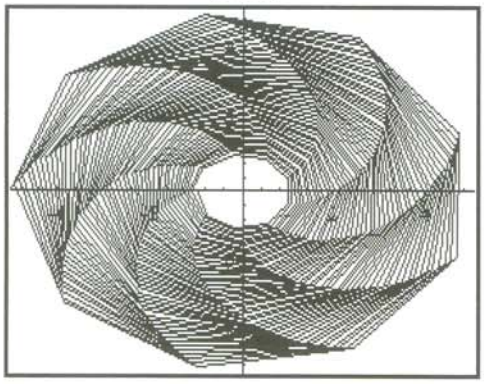
For comparison purposes, we have included in (D.3.9) through (D.3.12) four *Mathematica* generated graphs of this set of parametric equations. Figure (D.3.9) is the default graph obtained with the *Mathematica* command

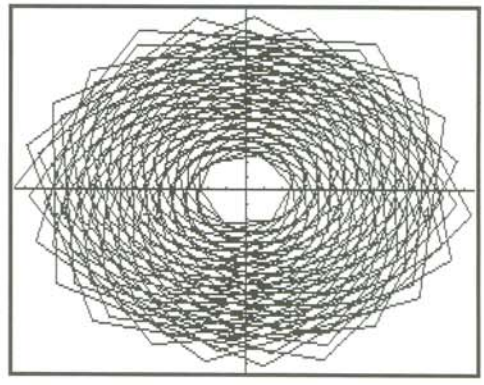
ParametricPlot[{t*Cos[25π*t],t*Sin[25π*t]},{t,1,4},
PlotRange->{{-5,5},{-5,5}}, Frame->False, PlotPoints->25].

Figures (D.3.10), (D.3.11), and (D.3.12) were obtained using the same command but with the option PlotPoints set to 20, 15, and 10, respectively. We should note that 25 is the default value for PlotPoints.


(D.3.9)


(D.3.10)


(D.3.11)


(D.3.12)

§4 – Calculus Example

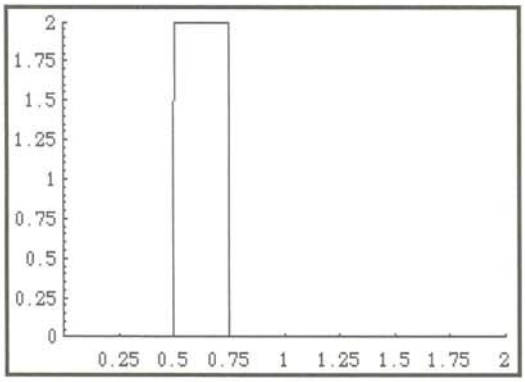
In §11 of Chapter 6, we used the **fnInt** built-in function to graph a function defined as an integral. But, as was seen in that section, even when used in graphing, **fnInt** can sometimes give erroneous results. For example, consider again the function

$$H(x) = \int_0^x h(t)\,dt,$$

where the integrand is the step function

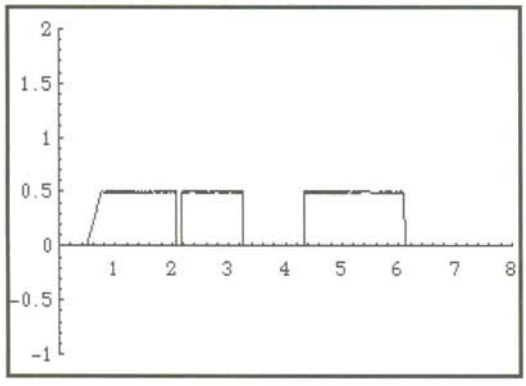
$$h(x) = \begin{cases} 2, & 0.5 \le x \le 0.75 \\ 0, & \text{elsewhere} \end{cases}.$$

Recall that the TI-86 did not correctly graph the function $H(x)$. Even *Mathematica* has difficulty graphing the $H(x)$ under consideration. Figures (D.4.1) and (D.4.2) show the *Mathematica* graphs for $h(x)$ and $H(x)$. Beside the two figures are the *Mathematica* commands used to get the graphs. It is true that *Mathematica* gives copious warnings about accuracy while it is graphing $H(x)$, but it is significant to notice how similar the *Mathematica* graph is to that of the TI-86 (see Figure 6.11.4).




```
Clear[g]
g[x_]:= If[x<.75 && x>.5,2,0];
Plot[g[x],{x,0,2},
  PlotRange->
                {{0,2},{0,2}}]
```

(D.4.1)




```
Plot[
NIntegrate[g[t],{t,0,x}],
  {x,0,8},
   PlotRange->
    {{0,8},{-1,2}},
   AxesOrigin->{0,0}]
```

(D.4.2)

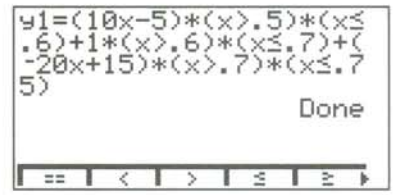
It is appropriate to discuss some details of the TI-86 integration algorithm. According to Texas Instruments documentation, the **fnInt** algorithm uses a (3,7) Gauss-Kronrod scheme. In this method, the value of the integral is approximated by a weighted average of function values of the integrand using three points to get a Gauss approximation G_3 and then seven points, three of which are the three used for G_3 but with different weights, to compute a Kronrod approximation K_7. If the difference between G_3 and K_7 is small enough, then the algorithm will quit. In particular, when the

TI-86 reported a value of 0 for $H(x)$ when $x \geq 0.75$, it is almost certainly the case that G_3 and K_7 both had value zero and the **fnInt** algorithm stopped prematurely. Another aspect of the Gauss-Kronrod method is that before any approximation actually takes place in computing a value for a definite integral, the integral is first transformed to one on the interval [–1, 1] by way of the transformation given by

$$\int_a^b f(t)\,dt = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}x + \frac{a+b}{2}\right)dx.$$

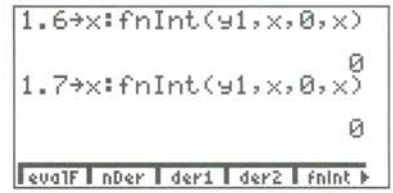
So it is clear that even a continuous piece-wise function whose only nonzero values are between 0.5 and 0.75 can be used in place of $h(x)$ and we will obtain results similar to our $H(x)$ function.

Figure (D.4.3) shows the defining of a function $y1$ on the home screen which is continuous but which is only nonzero between 0.5 and 0.75.



(D.4.3)

In (D.4.4), we compute $\int_0^x y1(t)\,dt$ for the values $x = 1.6$ and $x = 1.7$. Notice the similarity with (6.11.6) in Chapter 6.



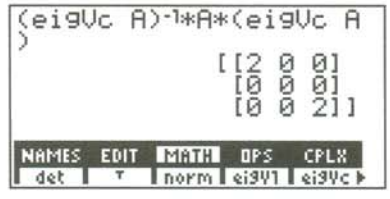
(D.4.4)

§5 – Matrix Example

In general, the algorithms used in doing a variety of matrix computations can be very sensitive to round-off error. This is certainly not unique to the TI-86, but it does lead to some interesting things which can be demonstrated with the help of the TI-86. For example, it is well known that an $n \times n$ matrix is similar to a diagonal matrix if, and only if, it has n linearly independent eigenvectors. The matrix **A** given by

$$\mathbf{A} = \begin{bmatrix} 2 & .00001 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 2 \end{bmatrix}$$

does not have three linearly independent eigenvectors, so it is not diagonalizable. But (D.5.1) seems to indicate otherwise. The matrix **eigVc A** should not be invertible, but round-off error has been enough of a factor that the TI-86 thinks it is dealing with a matrix whose columns are linearly independent.



(D.5.1)