

Chapter 1

Getting Started

1.1 Introduction

Ordinary differential equations (ODEs) are used throughout engineering, mathematics, and science to describe how physical quantities change, so an introductory course on elementary ODEs and their solutions is a standard part of the curriculum in these fields. Such a course provides insight, but the solution techniques discussed are generally unable to deal with the large, complicated, and nonlinear systems of equations seen in practice. This book is about solving ODEs numerically. Each of the authors has decades of experience in both industry and academia helping people like yourself solve problems. We begin in this chapter with a discussion of what is meant by a numerical solution with standard methods and, in particular, of what you can reasonably expect of standard software. In the chapters that follow, we discuss briefly the most popular methods for important classes of ODE problems. Examples are used throughout to show how to solve realistic problems. MATLAB (2000) is used to solve nearly all these problems because it is a very convenient and widely used *problem-solving environment* (PSE) with quality solvers that are exceptionally easy to use. It is also such a high-level programming language that programs are short, making it practical to list complete programs for all the examples. We also include some discussion of software available in other computing environments. Indeed, each of the authors has written ODE solvers widely used in general scientific computing.

An ODE represents a relationship between a function and its derivatives. One such relation taken up early in calculus courses is the linear ordinary differential equation

$$y'(t) = y(t) \tag{1.1}$$

which is to hold for, say, $0 \leq t \leq 10$. As we learn in a first course, we need more than just an ODE to specify a solution. Often solutions are specified by means of an initial value. For example, there is a unique solution of the ODE (1.1) for which $y(0) = 1$,

2 Chapter 1: Getting Started

namely $y(t) = e^t$. This is an example of an *initial value problem* (IVP) for an ODE. Like this example, the IVPs that arise in practice generally have one and only one solution. Sometimes solutions are specified in a more complicated way. This is important in practice, but it is not often discussed in a first course except possibly for the special case of Sturm–Liouville eigenproblems. Suppose that $y(x)$ satisfies the equation

$$y''(x) + y(x) = 0 \quad (1.2)$$

for $0 \leq x \leq b$. When a solution of this ODE is specified by conditions at both ends of the interval such as

$$y(0) = 0, \quad y(b) = 0$$

we speak of a *boundary value problem* (BVP). A Sturm–Liouville eigenproblem like this BVP always has the trivial solution $y(x) \equiv 0$, but for certain values of b there are non-trivial solutions, too. For instance, when $b = 2\pi$, the BVP has infinitely many solutions of the form $y(x) = \alpha \sin(x)$ for any constant α . In contrast to IVPs, which usually have a unique solution, the BVPs that arise in practice may have no solution, a unique solution, or more than one solution. If there is more than one solution, there may be a finite number or an infinite number of them.

Equation (1.1) tells us that the rate of change of the solution at time t is equal to the value of the solution then. In many physical situations, the effects of changes to the solution are delayed until a later time. Models of this behavior lead to *delay differential equations* (DDEs). Often the delays are taken to be constant. For example, if the situation modeled by the ODE (1.1) is such that the effect of a change in the solution is delayed by one time unit, then the DDE is

$$y'(t) = y(t - 1) \quad (1.3)$$

for, say, $0 \leq t \leq 10$. This problem resembles an initial value problem for an ODE; when the delays are constant, both the theory of DDEs and their numerical solution can be based on corresponding results for ODEs. There are, however, important differences. For the ODE (1.1), the initial value $y(0) = 1$ is enough to determine the solution, but that cannot be enough for the DDE (1.3). After all, when $t = 0$ we need $y(-1)$ to define $y'(0)$, but this is a value of the solution prior to the initial time. Thus, an initial value problem for the DDE (1.3) involves not just the value of the solution at the starting time but also its *history*. For this example it is easy enough to argue that, if we specify $y(t)$ for $-1 \leq t \leq 0$, then the initial value problem has a unique solution.

This book is about solving initial value problems for ODEs, boundary value problems for ODEs, and initial value problems for a class of DDEs with constant delays. For brevity we refer throughout to these three kinds of problems as IVPs, BVPs, and DDEs. In the rest of this chapter we discuss fundamental issues that are common to all three. Indeed,

some are so fundamental that – even if all you want is a little help solving a specific problem – you need to understand them. The IVPs are taken up in Chapter 2, BVPs in Chapter 3, and DDEs in Chapter 4. The IVP chapter comes first because the ideas and the software of that chapter are used later in the book, so some understanding of this material is needed to appreciate the chapters that follow. The chapters on BVPs and DDEs are mutually independent.

It is assumed that you are acquainted with the elements of programming in MATLAB, so we discuss only matters connected with solving ODEs. If you need to supplement your understanding of the language, the PSE itself has good documentation and there are a number of books available that provide more detail. One that we particularly like is the *MATLAB Guide* (Higham & Higham 2000). Most of the programs supplied with *Solving ODEs with MATLAB* plot solutions on the screen in color. Because it was not practical to provide color figures in the book, we modified the output of these programs to show the solutions in monochrome. Version 6.5 (Release 13) of MATLAB is required for Chapter 4, but version 6.1 suffices for the other chapters. Much of the cited software for general scientific computing is available from general-purpose, scientific computing libraries such as NAG (2002), Visual Numerics (IMSL 2002), and Harwell 2000 (*H2KL*), or from the Netlib Repository (*Netlib*). If the source of the software is not immediately obvious, it can be located through the classification system GAMS, the Guide to Available Mathematical Software (*GAMS*).

Numerical methods and the analytical tools of classical applied mathematics are complementary techniques for investigating and undertaking the solution of mathematical problems. You might be able to solve analytically simple equations involving a very few unknowns, especially with the assistance of a PSE for computer algebra like Maple (1998) or Mathematica (Wolfram 1996). All our examples were computed using the Maple kernel provided with the student version of MATLAB or using the Symbolic Toolbox provided with the professional version.

First we observe that even small changes to the equations can complicate greatly the analytical solutions. For example, Maple is used via MATLAB to solve the ODE

$$y' = y^2$$

at the command line by

```
>> y = dsolve('Dy = y^2')
```

```
y = -1/(t-C1)
```

(Sometimes we edit output slightly to give a more compact display.) In this general solution C1 is an arbitrary constant. This family of solutions expressed in terms of a familiar function gives us a lot of insight about how solutions behave. If the ODE is changed “slightly” to

4 Chapter 1: Getting Started

$$y' = y^2 + 1$$

then the general solution is found by `dsolve` to be

$$y = \tan(t+C1)$$

This is more complicated because it expresses the solution in terms of a special function, but it is at least a familiar special function and we understand well how it behaves. However, if the ODE is changed to

$$y' = y^2 + t$$

then the general solution found by `dsolve` is

$$y = (C1*\text{AiryAi}(1, -t) + \text{AiryBi}(1, -t)) / (C1*\text{AiryAi}(-t) + \text{AiryBi}(-t))$$

which in standard mathematical notation is

$$y(t) = \frac{C_1 Ai'(-t) + Bi'(-t)}{C_1 Ai(-t) + Bi(-t)}$$

Here $Ai(t)$ and $Bi(t)$ are Airy functions. (The Maple kernel denotes these functions by `AiryAi` and `AiryBi`, cf. `mhelp airy`; but MATLAB itself uses different names, cf. `help airy`.) Again C_1 is an arbitrary constant. The Airy functions are not so familiar. This solution is useful for studying the behavior of solutions analytically, but we'd need to plot some solutions to gain a sense of how they behave. Changing the ODE to

$$y' = y^2 + t^2$$

changes the general solution found by `dsolve` to

$$y = -t * (C1*\text{besselj}(-3/4, 1/2*t^2) + \text{bessely}(-3/4, 1/2*t^2)) / (C1*\text{besselj}(1/4, 1/2*t^2) + \text{bessely}(1/4, 1/2*t^2))$$

which in standard mathematical notation is

$$y(t) = -t \frac{C_1 J_{-3/4}\left(\frac{t^2}{2}\right) + Y_{-3/4}\left(\frac{t^2}{2}\right)}{C_1 J_{1/4}\left(\frac{t^2}{2}\right) + Y_{1/4}\left(\frac{t^2}{2}\right)}$$

Again the solution is expressed in terms of special functions, but now they are Bessel functions of fractional order. Again, we'd need to plot some solutions to gain insight. These equations are taken up later in Example 2.3.1.

Something different happens if we change the power of y :

```
>> y = dsolve('Dy = y^3 + t^2')  
Warning: Explicit solution could not be found.
```

This example shows that even simple-looking equations may not have a solution $y(t)$ that can be expressed in terms of familiar functions by Maple. Such examples are not rare, and usually when Maple fails to find an explicit solution it is because none is known. In fact, for a system of ODEs it is rare that an explicit solution can be found.

For these scalar ODEs it was easy to use a computer algebra package to obtain analytical solutions. Let us now consider some of the differences between solving ODEs analytically and numerically. The analytical solutions of the examples provide valuable insight, but to understand them better we'd need to evaluate and plot some particular solutions. For this we'd need to turn to numerical schemes for evaluating the special functions. But if we must use numerical methods for this, why bother solving them analytically at all? A direct numerical solution might be the best way to proceed for a particular IVP, but Airy and Bessel functions incorporate behavior that can be difficult for numerical methods to reproduce – namely, some have singularities and some oscillate very rapidly. If this is true of the solution that interests us or if we are interested in the solution as $t \rightarrow \infty$, then we may not be able to compute the solution numerically in a straightforward way. In effect, the analytical solution isolates the difficulties and we then rely upon the quality of the software for evaluating the special functions to compute an accurate solution. As the examples show, small changes to the ODE can lead to significant changes in the form of the analytical solution, though this may not imply that the behavior of the solution itself changes much. In contrast, there is really no difference solving IVPs numerically for these equations, including the one for which `dsolve` did not produce a solution. This illustrates the most important virtue of numerical methods: they make it easy to solve a large class of problems. Indeed, our considerable experience is that if an IVP arises in a practical situation, most likely you will not be able to solve it analytically yet you will be able to solve it numerically. On the other hand, the analytical solutions of the examples show how they depend on an arbitrary constant C_1 . Because numerical methods solve one problem at a time, it is not easy to determine how solutions depend on parameters. Such insight can be obtained by combining numerical methods with analytical tools such as variational equations and perturbation methods. Another difference between analytical and numerical solutions is that the standard numerical methods of this book apply only to ODEs defined by smooth functions that are to be solved on a finite interval. It is not unusual for physical problems to involve singular points or an infinite interval. Asymptotic expansions are often combined with numerical methods to deal with these difficulties.

In our view, analytical and numerical methods are complementary approaches to solving ODEs. This book is about numerical methods because they are easy to use and broadly applicable, but some kinds of difficulties can be resolved or understood only by analytical

6 Chapter 1: Getting Started

means. As a consequence, the chapters that follow feature many examples of using applied mathematics (e.g., asymptotic expansions and perturbation methods) to assist in the numerical solution of ODEs.

1.2 Existence, Uniqueness, and Well-Posedness

From the title of this section you might imagine that this is just another example of mathematicians being fussy. But it is not: it is about whether you will be able to solve a problem at all and, if you can, how well. In this book we'll see a good many examples of physical problems that do not have solutions for certain values of parameters. We'll also see physical problems that have more than one solution. Clearly we'll have trouble computing a solution that does not exist, and if there is more than one solution then we'll have trouble computing the "right" one. Although there are mathematical results that guarantee a problem has a solution and only one, there is no substitute for an understanding of the phenomena being modeled.

Existence and uniqueness are much simpler for IVPs than BVPs, and the class of DDEs we consider can be understood in terms of IVPs, so we concentrate here on IVPs and defer to later chapters a fuller discussion of BVPs and DDEs. The vast majority of IVPs that arise in practice can be written as a system of d explicit first-order ODEs:

$$\begin{aligned}y_1'(t) &= f_1(t, y_1(t), y_2(t), \dots, y_d(t)) \\y_2'(t) &= f_2(t, y_1(t), y_2(t), \dots, y_d(t)) \\&\vdots \\y_d'(t) &= f_d(t, y_1(t), y_2(t), \dots, y_d(t))\end{aligned}$$

For brevity we generally write this system in terms of the (column) vectors

$$y(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_d(t) \end{pmatrix}, \quad f(t, y(t)) = \begin{pmatrix} f_1(t, y(t)) \\ f_2(t, y(t)) \\ \vdots \\ f_d(t, y(t)) \end{pmatrix}$$

as

$$y'(t) = f(t, y(t)) \tag{1.4}$$

An IVP is specified by giving values of all the solution components at an initial point,

$$y_1(a) = A_1, y_2(a) = A_2, \dots, y_d(a) = A_d$$

1.2 Existence, Uniqueness, and Well-Posedness

7

or, in vector notation,

$$y(a) = A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_d \end{pmatrix} \quad (1.5)$$

Using vectors, a system of first-order equations resembles a single equation; in fact, the theory is much the same. However, writing problems as first-order systems is not only convenient for the theory, it is critically important in practice. We'll explain this later and show how to do it.

Roughly speaking, if the function $f(t, y)$ is smooth for all values (t, y) in a region R that contains the initial data (a, A) , then the IVP comprising the ODE (1.4) and the initial condition (1.5) has a solution and only one. This settles the existence and uniqueness question for most of the IVPs that arise in practice, but we need to expand on the issue of where the solution exists. The solution extends to the boundary of the region R , but that is not the same as saying that it exists throughout a given interval $a \leq t \leq b$ contained in the region R . An example makes the point. The IVP

$$y' = y^2, \quad y(0) = 1$$

has a function $f(t, y) = y^2$ that is smooth everywhere; in other words, it is smooth in the region

$$R = \{-\infty < t < \infty, -\infty < y < \infty\}$$

Yet the unique solution

$$y(t) = \frac{1}{1-t}$$

“blows up” as $t \rightarrow 1$ and hence does not exist on a whole interval $0 \leq t \leq 2$ (say) that is entirely contained in R . This does not contradict the existence result because as $t \rightarrow 1$, the solution approaches the boundary of the region R in the y variable, a boundary that happens to be at infinity. This kind of behavior is not at all unusual for physical problems. Correspondingly, it is usually reasonable to ask that a numerical scheme approximate a solution well until it becomes too large for the arithmetic of the computer used. Exercises 1.2 and 1.3 take up similar cases.

The form of the ODEs (1.4) and the initial condition (1.5) is standard for IVPs, and in Section 1.3 we look at some examples showing how to write problems in this form. Existence and uniqueness is relatively simple for this standard explicit form, but the properties are more difficult to analyze for equations in the implicit form

$$F(t, y(t), y'(t)) = 0$$

8 Chapter 1: Getting Started

Very simple examples show that both existence and uniqueness are problematic for such equations. For instance, the equation

$$(y'(t))^2 + 1 = 0$$

obviously has no (real) solutions. A more substantial example helps make the point. In scientific and engineering applications there is a great deal of interest in how the solutions y of a system of algebraic equations

$$F(y, \lambda) = 0$$

depend on a (scalar) parameter λ . Differentiating with respect to the parameter, we find that

$$\frac{\partial F}{\partial y} \frac{dy}{d\lambda} + \frac{\partial F}{\partial \lambda} = 0$$

This is a system of first-order ODEs. If for some λ_0 we can solve the algebraic equations $F(y, \lambda_0) = 0$ for $y(\lambda_0) = y_0$, then this provides an initial condition for an IVP for $y(\lambda)$. If the Jacobian matrix

$$J = \frac{\partial F}{\partial y} = \left(\frac{\partial F_i}{\partial y_j} \right)$$

is nonsingular, we can write the ODEs in the standard form

$$\frac{dy}{d\lambda} = -J^{-1} \frac{\partial F}{\partial \lambda}$$

However, if the Jacobian matrix is singular then the questions of existence and uniqueness are much more difficult to answer. This is a rather special situation, but in fact it is often the situation with the most interesting science. It is when solutions bifurcate – that is, the number of solutions changes. If we are to apply standard codes for IVPs at such a singular (bifurcation) point, we must resort to the analytical tools of applied mathematics to sort out the behavior of solutions near this point. Exercise 1.1 considers a similar problem.

As a concrete example of bifurcation, suppose that we are interested in steady-state (constant) solutions of the ODE

$$y' = y^2 - \lambda$$

The steady states are solutions of the algebraic equation

$$0 = y^2 - \lambda \equiv F(y, \lambda)$$

It is obvious that, for $\lambda \geq 0$, one steady-state solution is $y(\lambda) = \sqrt{\lambda}$. However, to study more generally how the steady state depends on λ , we could compute it as the solution of the IVP

1.2 Existence, Uniqueness, and Well-Posedness

9

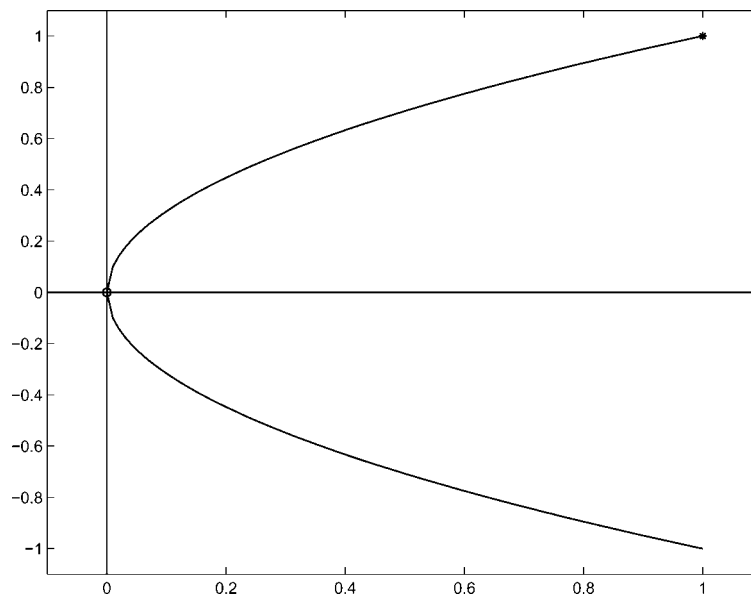


Figure 1.1: $(0, 0)$ is a singular point for $2yy' - 1 = 0$.

$$2y \frac{dy}{d\lambda} - 1 = 0, \quad y(1) = 1$$

Provided that $y \neq 0$, the ODE can be written immediately in standard form and solved for values of λ decreasing from 1. However, the equation is singular when $y(\lambda) = 0$, which is true for $\lambda = 0$. The singular point $(0, 0)$ leaves open the possibility that there is more than one solution of the ODE passing through this point, and so there is: $y(\lambda) = -\sqrt{\lambda}$ is a second solution. Using standard software, we can start at $\lambda = 1$ and integrate the equation easily until close to the origin, where we run into trouble because $y'(\lambda) \rightarrow \infty$ as $\lambda \rightarrow 0$. See Figure 1.1.

For later use in discussing numerical methods, we need to be a little more precise about what we mean by a *smooth* function $f(t, y)$. We mean that it is continuous in a region R and that it has continuous derivatives with respect to the dependent variables there – as many derivatives as necessary for whatever argument we make. A technical condition is that f must satisfy a *Lipschitz condition* in the region R . That is, there is a constant L such that, for any points (t, u) and (t, v) in the region R ,

$$\|f(t, u) - f(t, v)\| \leq L\|u - v\|$$

In the case of a single equation, the mean value theorem states that

10 Chapter 1: Getting Started

$$f(t, u) - f(t, v) = \frac{\partial f}{\partial y}(t, \zeta)(u - v)$$

so $f(t, y)$ satisfies a Lipschitz condition if $\left| \frac{\partial f(t, y)}{\partial y} \right|$ is bounded in the region R by a constant L . Similarly, if the first partial derivatives $\left| \frac{\partial f_i(t, y_1, y_2, \dots, y_d)}{\partial y_j} \right|$ are all bounded in the region R , then the vector function $f(t, y)$ satisfies a Lipschitz condition there.

Roughly speaking, a *well-posed problem* is one for which small changes to the data lead to small changes in the solution. Such a problem is also said to be *well-conditioned* with respect to changes in the data. This is a fundamental property of a physical problem and it is also fundamental to the numerical solution of the problem. The methods that we study can be regarded as producing the exact solution to a problem with the data that defines the problem changed a little. For a well-posed problem, this means that the numerical solution is close to the solution of the given problem. In practice this is all blurred because it depends both on how much accuracy you want in a solution and on the arithmetic you use in computing it. Let's now discuss a familiar example that illuminates some of the issues.

Imagine that we have a pendulum: a light, rigid rod hanging vertically from a frictionless pivot with a heavy weight (the bob) at the free end. With a particular choice of units, the angle $\theta(t)$ that the pendulum makes with the vertical at time t satisfies the ODE

$$\theta'' + \sin(\theta) = 0 \tag{1.6}$$

Suppose that the pendulum is hanging vertically so that the initial angle $\theta(0) = 0$ and that we thump the bob to give it an initial velocity $\theta'(0)$. When the initial velocity is zero, the pendulum does not move at all. If the velocity is nonzero and small enough, the pendulum will swing back and forth. Figure 1.2 shows $\theta(t)$ for several such solutions, namely those with initial velocities $\theta'(0) = -1.9, 1.5, \text{ and } 1.9$. There is another kind of solution. If we thump the bob hard enough, the pendulum will swing over the top and, with no friction, it will whirl around the pivot forever. This is to say that if the initial velocity $\theta'(0)$ is large enough then $\theta(t)$ will increase forever. The figure shows two such solutions with initial velocities $\theta'(0) = 2.1$ and 2.5 . If you think about it, you'll realize that there is a very special solution that occurs as the solutions change from oscillatory to increasing. This solution is the dotted curve in Figure 1.2. Physically, it corresponds to an initial velocity that causes the pendulum to approach and then come to rest vertically and upside down. Clearly this solution is unstable – an arbitrarily small change to the initial velocity gives rise to a solution that is eventually very different. In other words, the IVP for this initial velocity is ill-posed (ill-conditioned) on long time intervals.

Interestingly, we can deduce the initial velocity that results in the unstable solution of (1.6). This is a conservative system, meaning that the *energy*

$$E(t) = 0.5(\theta'(t))^2 - \cos(\theta(t))$$