

# 1. Introduction

What if all you had to do to solve an ODE were just to write it down?<sup>1</sup> That is the line we will follow in this book. Our emphasis is not just on the mathematics of ODEs, but on how the solutions behave. Do they blow up, decay, oscillate? Are there rapid transitions where they flip from one state to another? Does the behavior change if a coefficient is perturbed or a new term is added? And how can such variety be deployed to explain the world around us? We shall not just talk about these matters but explore them in action.

ODEs are among the core topics of mathematics, with applications so ubiquitous that listing examples almost seems inappropriate. (Heat conduction, chemical reactions, chaos, population dynamics, deformations of a beam, radioactivity, bifurcation theory, stability theory, differential geometry, quantum mechanics, economics, finance, infectious diseases, nerve signals, vibrations, optics, waves, dynamics of networks, special functions, ballistics, planetary dynamics, . . . .) ODEs are everywhere.

To solve ODEs by writing them down, we will use Chebfun, an open-source MATLAB package that is freely available at [www.chebfun.org](http://www.chebfun.org). In MATLAB, you type  $\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$  to solve the system of equations  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , where  $\mathbf{A}$  is a matrix and  $\mathbf{x}$  and  $\mathbf{b}$  are vectors. In Chebfun, analogously, you type  $y = L \backslash f$  to solve the ODE  $Ly = f$ , where  $L$  is a linear or nonlinear differential operator with initial or boundary conditions and  $y$  and  $f$  are functions. We will obtain solutions this way on nearly every page, presenting them with hundreds of computer-generated plots without discussing the algorithms Chebfun uses to make this

---

<sup>1</sup>ODE stands for *ordinary differential equation*, as the reader presumably knows already.

possible.<sup>2</sup> This is not a book about numerical analysis or computer science. It is a book about ODEs.

Many textbooks on ODEs concentrate on linear problems, because nonlinear ones are rarely analytically solvable. Here, with analytical solutions playing a lesser role, we will be able to give a more balanced treatment and fully appreciate the remarkable effects that come with nonlinearity.

Let's get started with the most basic **initial-value problem (IVP)**,<sup>3</sup>

$$y' + y = 0, \quad t \in [0, 3], \quad y(0) = 1. \quad (1.1)$$

We have written this in the standard notation of this book, with  $t$  as the independent variable,  $y$  the dependent variable, and  $y' = dy/dt$ .<sup>4</sup> Note that although  $y$  and  $y'$  are functions of  $t$ , we usually do not write them out fully as  $y(t)$  and  $y'(t)$ . You can quickly check that a solution to (1.1) is the function  $y(t) = \exp(-t)$ , and moreover, it is easily proved that this solution is unique (see Chapter 2). To calculate it with Chebfun, we make a “chebop” that encodes the differential operator of (1.1), which we call  $L$ . First we prescribe the interval.

```
L = chebop(0,3);
```

Next we prescribe the differential operator  $y \mapsto y' + y$ , which is written in the form of a MATLAB anonymous function of  $t$  and  $y$ , with `diff` denoting differentiation with respect to  $t$ .

```
L.op = @(t,y) diff(y) + y;
```

Finally we set the *initial condition* with `L.lbc`, which stands for “left boundary condition.”

```
L.lbc = 1;
```

(A boundary condition at the right would be specified with `L.rbc`.) We can now solve (1.1).

```
y = L\0;
```

As expected, the solution is  $e^{-t}$ .<sup>5</sup>

---

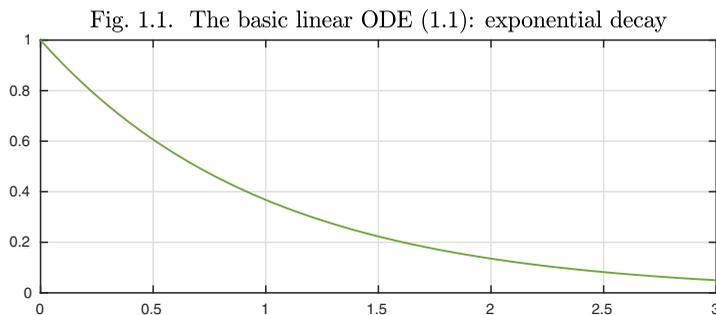
<sup>2</sup>To learn about Chebfun and its ODE algorithms, see Appendix A and [www.chebfun.org](http://www.chebfun.org).

<sup>3</sup>Throughout this book, some terms are set in italics, while other particularly important ones are set in boldface. To review a chapter, a good way to start is to read the items in italics and boldface and also the chapter summary at the end. Almost all emphasized terms can be found in the index.

<sup>4</sup>For boundary-value problems, which are usually associated with space instead of time, we will change  $t$  to  $x$ . Some books also make a distinction between  $y'$  for a space derivative and  $\dot{y}$  for a time derivative, but we shall not do this.

<sup>5</sup>Throughout this book, the Chebfun code segments listed are sufficient to reproduce the mathematical essence of each figure, but MATLAB formatting commands like `title`, `axis`, and `'linewidth'` have been removed. Users wanting to see formatting details can download the M-files for each chapter of the book from [www.chebfun.org](http://www.chebfun.org).

plot(y)



The problem we have just solved can be classified by the following properties.

- F First-order** (the highest-order derivative is  $y'$ )
- L Linear** (there are no terms like  $y^2$ ,  $\exp(y)$ , or  $yy'$ )
- A Autonomous** (the equation, though not the solution, is independent of  $t$ )
- S Scalar** (there is just one dependent variable  $y$  rather than  $u, v, w, \dots$ )
- H Homogeneous** (the right-hand side is zero<sup>6</sup>)
- I Initial-value problem** (not a boundary-value or eigenvalue problem)

Note that these letters spell the easily remembered word **FLASHI**. In this book, we will use this word occasionally to encode some of the properties of a problem. If we state that a problem is, say, of type **FlaShi**, that means that it is a first-order scalar problem, since the letters **F** and **S** are capitalized. On the other hand since **l**, **a**, **h**, and **i** are in lowercase, the problem is nonlinear, nonautonomous, inhomogeneous, and a boundary-value or eigenvalue problem rather than an IVP. In the chapters ahead, we will vary all of these properties and explore in the process a great variety of phenomena. The **FLASHI** classification will help add structure to the discussion. Sometimes one considers an ODE without specifying initial or boundary conditions, and in this case the **I** drops away to give just **FLASH**.

Here is a **FLASH** classification of the theorems in this book.

*Linear, first-order, scalar*

**FLASH**: Thm. 2.1 (separation of variables)

**FLASh**: Thm. 2.4 (integrating factor)

**FLaSH**: Thm. 2.2 (separation of variables)

**FLaSh**: Thm. 2.3 (integrating factor), Thm. 19.1 (periodic)

*Nonlinear, first-order, scalar*

**FIASH**: Thm. 3.2 (separation of variables)

---

<sup>6</sup>When we say “the right-hand side is zero,” we really mean that the equation contains no nonzero terms that do not involve  $y$ . Of course it doesn’t matter mathematically whether a term appears on the left or the right of an equation. For nonlinear ODEs, the property of homogeneity does not always have much meaning.

FlaSH: Thm. 3.1 (separation of variables)

FlaSh: Thms. 11.1 and 11.3 (Picard iteration)

*Linear, first-order, system*

FLAsH: Thm. 14.1 (matrix exponential)

FLAsh: Thm. 14.2 (matrix exponential), Thm. 14.3 (variation of parameters)

FLash: Thm. 19.2 (periodic)

*Nonlinear, first-order, system*

FIAsH: Thm. 14.4 (linearization), Thm. 15.1 (stability)

Flash: Thms. 11.2 and 11.3 (Picard iteration)

*Second-order*

fLASH: Thm. 4.1 (solution formula)

fLaSH: Thm. 7.1 (eigenproblems), Thm. 19.3 (Hill's eq.)

fIASH: Thm. 14.5 (linearization)

As an illustration of a nonlinear ODE, here is an example of type fIASHI, the second-order equation known as the *van der Pol equation*:

$$0.3y'' - (1 - y^2)y' + y = 0, \quad t \in [0, 20], \quad y(0) = 1, \quad y'(0) = 0. \quad (1.2)$$

(The coefficient 0.3 is included to make the solution more interesting.) Because  $y''$  is present, the equation is of second rather than first order (hence f not F), and it is nonlinear because of the coefficient  $1 - y^2$  multiplying  $y'$  (hence l not L). Here are the interval and the operator, which we name N instead of L as a reminder that it is nonlinear.

```
N = chebop(0,20);
```

```
N.op = @(t,y) 0.3*diff(y,2) - (1-y^2)*diff(y) + y;
```

Problem (1.1) had just a single boundary condition, but in (1.2), since it is a second-order equation, there are two. For a simple scalar problem like this, Chebfun permits one to prescribe  $y$  and  $y'$  at a point by supplying a vector of two numbers.

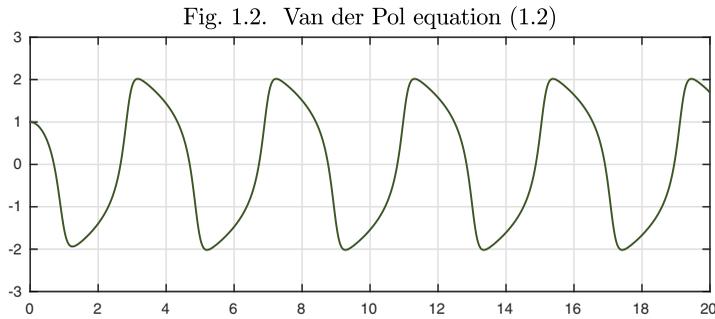
```
N.lbc = [1;0];
```

Here is the solution.<sup>7</sup>

```
y = N\0; plot(y)
```

---

<sup>7</sup>Readers viewing these pages in color will note that solutions to linear IVPs are usually plotted in green and solutions to nonlinear ones in dark green. Starting in Chapter 5, we will likewise plot solutions to linear BVPs in blue and solutions to nonlinear ones in dark blue. The distinctions between linear and nonlinear equations and between IVPs and BVPs are important, and the colors will serve as a quiet reminder.



Chebfun gives us ready access to the properties of a computed solution  $y$  (which is represented as a “chebfun,” with a lowercase  $c$  since this is a function rather than the name of the software system). For example, here are the positions of the local maxima of  $y$ .

```
[mval,mpos] = max(y,'local'); mpos'
```

```
ans =
```

```
0    3.1651    7.2375   11.3101   15.3826   19.4552
```

By taking differences of successive maxima, we see that this van der Pol oscillation is settling down to a periodic function with period 4.0725,

```
diff(mpos)'
```

```
ans =
```

```
3.1651    4.0725    4.0725    4.0725    4.0725
```

This book is aimed at everyone who is interested in ODEs. If you are an undergraduate taking a course from one of the big texts like Boyce and DiPrima or Edwards and Penney, this is your lightweight companion. (The hard copy from SIAM is inexpensive, and the online version is free.) If you are a graduate student working in any of the mathematical sciences, this may be just the book to take your understanding to the next level. Whoever you are, we aim to increase your appreciation of this fundamental subject.

What does it mean to “solve” an ODE? One kind of solution would be an exact explicit formula, also known as an *analytical solution*. Advantages of analytical solutions include perfect accuracy, generality, explicit dependence on parameters, theoretical insight, and the absence of restriction to a particular range of values of  $t$ . The trouble is, most ODE problems, including almost all nonlinear ones, can’t be solved analytically. Another kind of solution is a numerical one obtained on a computer. The great advantage of numerical solutions is that they can be obtained for virtually any ODE. That’s not the only advantage, however. Another is that numbers are a compact and universal currency, so that by examining results obtained numerically, one can apply one’s analysis to a problem at hand or check how one ODE solution compares to another. It is not always obvious how to compare two exact formulas, but

we always know how to compare two numbers, whether explicitly as numbers or visually in a plot.

Among the huge variety of interesting ODEs, there are five that keep reappearing over and over in this book:

**Second-order linear oscillator.** Simple harmonic motion, or with damping.

**Van der Pol equation.** Nonlinear oscillator, with solutions on a limit cycle.

**Nonlinear pendulum.** Large-amplitude, with a periodic phase space.

**Lorenz equations.** Archetypal chaotic system, with three coupled variables.

**Linear system.**  $\mathbf{y}' = \mathbf{A}\mathbf{y}$ , with solutions  $\exp(t\mathbf{A})\mathbf{y}_0$ .

Many other equations will also be mentioned, including the Airy, Belousov–Zhabotinsky, Bernoulli, Bessel, Blasius, Bratu, Brusselator, Carrier, Duffing, Hénon–Heiles, Hill, logistic, Lotka–Volterra, Mathieu,  $n$ -body, Oregonator, Painlevé, and Rössler equations.

Easy computer exploration brings new perspectives on ODEs, and we believe we have found distinctive treatments of most of the topics presented in this book, which we hope will blend clarity for beginners with unexpected insights for experts. Here are some items worthy of note in each chapter.

1. *Introduction.* The “FLASHI” classification.
2. *First-order scalar linear ODEs.* Fig. 2.3: smooth vs. bang-bang forcing.
3. *First-order scalar nonlinear ODEs.* Scalarization by complex arithmetic.
4. *Second-order ODEs and damping.* Ex. 4.1: elliptical/nonelliptical orbits.
5. *Boundary-value problems.* Fig. 5.9: side conditions.
6. *Eigenvalues of linear BVPs.* Fig. 6.2: eigenvalues via response curve.
7. *Variable coefficients and adjoints.* Automated computation of adjoints.
8. *Resonance.* Periodic forcing gives periodic solutions, *if* there is damping.
9. *Second-order equations in the phase plane.* BVPs as well as IVPs.
10. *Systems of equations.* SIR epidemiology models in 6 lines of code.
11. *The fundamental existence theorem.* Picard iteration on the computer.
12. *Random functions and random ODEs.* Stochastics via smooth functions.
13. *Chaos.* Transient chaos in the 3-body problem.
14. *Linearization.* Figures showing that, locally, any ODE behaves linearly.
15. *Stable and unstable fixed points.* Application to transition to turbulence.
16. *Multiple solutions of nonlinear BVPs.* Shooting to find multiple solutions.
17. *Bifurcation.* Fig. 17.18: tracking hysteresis as a parameter varies.
18. *Continuation and path-following.* Fig. 18.13: numerical bifurcation.
19. *Periodic ODEs.* Application showing the origin of band gaps.
20. *Boundary and interior layers.* Plots showing asymptotics in action.
21. *Into the complex plane.* Analytic continuation of real solutions.
22. *Time-dependent PDEs.* How ODEs arise from PDEs as  $t \rightarrow \infty$ .

Readers will also find many phenomena explored in the exercises that do not appear in other textbooks.

Any book on ODEs faces the question of how much space to give to applications. As a structure that we hope will prove appealing, we follow the pattern that each chapter ends with a 2–4 page item designated as an Application. This

is followed in turn by a few sentences about history, a mention of our favorite reference, and the exercises for that chapter.

With the help of Chebfun, can we really solve any ODE just by writing it down? No, of course not. Examples can readily be devised that defeat this method for reasons including singularities, scaling, stiffness, positivity constraints, or sheer computational scale, and a page about such challenges can be found at the end of Appendix A. Nevertheless, we have been gratified in writing this book to find how easy it has been to explore almost any topic.

---

#### APPLICATIONS IN THIS BOOK

---

- Chapter 2.* Elimination of caffeine from the bloodstream.
- Chapter 3.* Classic pursuit problems.
- Chapter 4.* Skydiver.
- Chapter 5.* Beam theory and the strength of spaghetti.
- Chapter 6.* Eigenstates of the Schrödinger equation.
- Chapter 7.* Adjoint and optimization.
- Chapter 8.* Moon, sun, and tides.
- Chapter 9.* Nonlinear pendulum.
- Chapter 10.* SIR model for epidemics.
- Chapter 11.* Designer nonuniqueness.
- Chapter 12.* Metastability, radioactivity, and tunneling.
- Chapter 13.* Chaos in a food web.
- Chapter 14.* Linearized Lorenz trajectories.
- Chapter 15.* Transition to turbulence in a pipe.
- Chapter 16.* Sending a spacecraft to a destination.
- Chapter 17.* FitzHugh–Nagumo equations of neural signals.
- Chapter 18.* Arrhenius chemical reaction.
- Chapter 19.* Band gaps and forbidden frequencies.
- Chapter 20.* Why is New York hotter than San Francisco?
- Chapter 21.* Jacobi sine function.
- Chapter 22.* Solitons and the KdV equation.

**HISTORY.** Many of the great mathematicians of the past were involved in establishing the subject of ODEs, starting around 1670, including Newton, Leibniz, Johann and Jacob Bernoulli, Riccati, Clairaut, Euler, d’Alembert, Lagrange, Gauss, and Cauchy. Stepping forward to 100 or so years ago, some other key figures were Poincaré, Picard, Lyapunov, Painlevé, and Goursat.

**OUR FAVORITE REFERENCE.** For a charismatic tour of the whole subject of ODEs with a historical emphasis, see Chapter I of Hairer, Nørsett, and Wanner, *Solving Ordinary Differential Equations I*, Second Revised Edition, Springer-Verlag, 1993.

**ACKNOWLEDGMENTS.** Many people have helped us in preparing this book, and we can acknowledge just a few: David Allwright, Patrick Farrell, Abdul-

Lateef Haji-Ali, Nick Hale, Des Higham, Hrothgar, Aurya Javeed, and Tadashi Tokieda. Michael Rawson was a lively participant in a course at NYU that class-tested the book in autumn 2016. We are particularly grateful to Abinand Gopal, Matt Meyers, Niels Møller, and Adam Stinchcombe for reading the entire manuscript at a late stage and offering many helpful suggestions.

SUMMARY OF CHAPTER 1. *ODEs can rarely be solved analytically, but they can always be solved numerically, and Chebfun provides a convenient tool for doing this. This book explores all kinds of ODE problems, from the elementary to the advanced, both initial-value problems (IVPs) and boundary-value problems (BVPs). ODE problems can be classified by a schema with mnemonic FLASHI: F = first-order, L = linear, A = autonomous, S = scalar, H = homogeneous, I = IVP.*

*A note on exercises.* The exercises in this book mix the theoretical and the computational, and the labels of computational exercises are underlined. If you are asked, say, to “find a value” of a solution, a computational result is usually expected unless it is explicitly stated that it should be analytical. As a rule, give computed results to 6 digits of accuracy. For theoretical problems, though your final solution should be analytical, you may sometimes find computational explorations helpful along the way.

Exercise 1.1. *Local extrema of van der Pol oscillation.* In the van der Pol example (1.2), you can find the local maxima with `max(y, 'local')`, and similarly for minima with `min`; you can find both minima and maxima at once with `minandmax`. How close is the first local minimum value (at  $t \approx 1.2$ ) to its asymptotic value for  $t \rightarrow \infty$ ? Likewise for the first local maximum (at  $t \approx 3.2$ )?

*Exercise 1.2. Classification of ODE problems.* Classify the following ODE problems according to the FLASHI scheme.

- (a)  $y' = \sin(t) - y$ ,  $t \in [0, 100]$ ,  $y(0) = 1$ .
- (b)  $y' = \sin(t) - y^3$ ,  $t \in [0, 100]$ ,  $y(0) = 1$ .
- (c) (Nonlinear pendulum equation)  $y'' = -\sin(y)$ ,  $t \in [0, 10]$ ,  $y(0) = y(10) = 2$ .
- (d) (Advection-diffusion equation)  $0.02y'' + y' + y = 0$ ,  $t \in [0, 1]$ ,  $y(0) = 0$ ,  $y(1) = 1$ .
- (e) (Airy equation)  $0.02y'' - ty = 0$ ,  $t \in [-5, 5]$ ,  $y(-5) = 1$ ,  $y(5) = 0$ .
- (f) (Harmonic oscillator)  $u' = v$ ,  $v' = -u$ ,  $t \in [0, 100]$ ,  $u(0) = 1$ ,  $v(0) = 0$ .
- (g)  $u' = u^2v$ ,  $v' = -uv^2$ ,  $t \in [0, 2]$ ,  $u(0) = 1$ ,  $v(0) = 0$ .
- (h) (Bessel equation)  $t^2y'' + ty' + (t^2 - 4)y = 0$ ,  $t \in [0, 8]$ ,  $y(0) = 0$ ,  $y(8) = 1$ .
- (i)  $0.1y'' + yy' = y$ ,  $t \in [-1, 1]$ ,  $y(-1) = -2$ ,  $y(1) = 1$ .
- (j) (Lotka–Volterra equations)  $u' = u(1 - v)$ ,  $v' = v(u - 1)$ ,  $t \in [0, 10]$ ,  $u(0) = v(0) = 1$ .
- (k) (Blasius equation)  $y''' + 0.5yy'' = 0$ ,  $t \in [0, 10]$ ,  $y(0) = y'(0) = 0$ ,  $y'(10) = 1$ .

Exercise 1.3. *Airy equation.* Plot the solution  $y$  of the problem of Exercise 1.2(e) and report its maximum value. Letting  $k$  denote the coefficient 0.02, do the same with  $k = 0.002$ . Now make a plot of  $\max(y)$  as a function of  $k$  for the values  $k = 0.001, 0.002, \dots, 0.039, 0.040$ . (We shall explore such effects in Chapter 6.)

Exercise 1.4. *Solution with rapid transient.* Plot the solution  $y$  of the problem of Exercise 1.2(i), and give its maximum slope  $s = \max_{t \in [-1, 1]} y'(t)$ . Do the same with the coefficient 0.1 reduced to  $1/20$ ,  $1/40$ , and  $1/80$ .

*Exercise 1.5. Reduction to first-order system.* (a) Show how the IVP of Exercise 1.2(f) can be rewritten as a second-order scalar IVP involving just the dependent variable  $u$ . What are the initial conditions for this IVP? (b) Conversely, show how the third-order IVP of Exercise 1.2(k) can be rewritten as a first-order system involving three variables  $u$ ,  $v$ , and  $w$ . Any higher-order ODE problem can be rewritten as a first-order problem like this, and numerical software for IVPs often requires the problem to be expressed in first-order form.

*Exercise 1.6. How Chebfun represents functions.* Chebfun normally represents solutions  $y(t)$  to ODEs by polynomial approximations, typically with an accuracy of about 10 digits, whose degree  $n$  may be quite high. The polynomials can be interpreted as interpolants through a sufficiently large number  $n + 1$  of samples at *Chebyshev points* defined by  $t_j = \cos(\pi j/n)$ ,  $0 \leq j \leq n$  for  $t \in [-1, 1]$ , or linearly transplanted to a different interval  $[a, b]$ . (a) Let  $\mathbf{y}$  be the computed solution of the problem (1.1). Execute `length(y)` to find the number  $n + 1$  for this function, and `plot(y, '-')` to see the associated Chebyshev points. (b) Do the same for the computed solution of the van der Pol problem (1.2). Approximately speaking (say, to within 10%), how many interpolation points are there on average per wavelength?

*Exercise 1.7. How Chebfun represents periodic functions.* Chebfun also has a representation for periodic functions that takes advantage of the periodicity, based on trigonometric polynomials (i.e., Fourier series) rather than ordinary algebraic polynomials. Periodic solutions arise naturally in ODEs with periodic coefficients (see Chapter 19). (a) Construct an ordinary chebfun for  $f(t) = (1.1 - \cos(\pi t))^{-1}$ ,  $t \in [-1, 1]$  with the command `chebfun('1/(1.1-cos(pi*x))')`. What is its length? (b) How does the length change if you use `chebfun('1/(1.1-cos(pi*x))','trig')`? (In an appropriate limit, the ratio of the two lengths approaches  $\pi/2$ .)