

Chapter 3

Nonlinearity in differential equations: models and numerical solutions

3.1 Introduction

In the first two chapters we found general solutions for linear dynamical systems, both in discrete and continuous time. Most interesting biological models are not linear, and do not have analytic solutions. Nonlinear differential equations are a rich and complex field of study for mathematicians, and the behavior of their solution remains poorly understood in many cases. For ODEs with one variable that are the topic of this unit, there are some results that guarantee the existence and uniqueness of these solutions for reasonable equations. However, this generally does not mean that these solutions are easy to find.

Biological models require nonlinearity in order to capture the interdependence of various factors. These models tend to exhibit complex dynamics, which often better describe the dynamics of the biological systems under consideration. But as was mentioned above, nonlinear differential equations may not be solvable even theoretically. For the practical modeler, when solutions cannot

be found on paper, computational methods come to the rescue. In this chapter we will encounter the simplest numerical methods, which allow a computer program to find an approximate solution of an ODE.

In this chapter, the modeling section presents examples of nonlinear models, with explanations of the origin of the nonlinear terms. The analytical section describes the theory of existence of solutions, and introduces the concept of finite difference schemes. The computational section presents the two simplest finite difference schemes, the Forward and Backward Euler, and defines fundamental concepts of stability and accuracy of numerical schemes. In the synthesis section, we analyze a model of ovarian follicle maturation, using the qualitative methods from Chapter 2 and the numerical schemes from the computational section.

3.2 Modeling: nonlinear model of infection

3.2.1 Sources of nonlinearity

In the first two chapters we saw linear ODEs describing population growth with constant rates, chemical kinetics with constant rates, and membrane potential with constant ion channel conductances. You see the pattern: linear models are applicable when the relationship in the ODE is independent of the actual value of the dependent variable. In our examples, this means that reproduction and death are not affected by population size, that reactions occur at the same rate regardless of concentration, and that ions flow through a channel with the same speed regardless of the voltage across the membrane. With the possible exception of the chemical kinetic example, these models do not match reality.

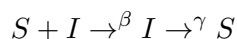
We saw how adding a simple dependence of population growth rate on the population size resulted in the nonlinear logistic model $\dot{x} = rx(k-x)$. This model can be re-written as $\dot{x} = ax - bx^2$, so it is clear that there is a *linear term* (ax) and a nonlinear term $-bx^2$. When x is sufficiently

small (and positive) the linear term is greater, and the population grows. When x is large enough, the nonlinear term wins and the population declines.

3.2.2 Simple epidemiology model

Let us consider a model of infectious disease in a (human) population. For this model, there are two variables to be tracked: the number of susceptible (S) and infected (I) individuals in the population. The susceptible individuals can get infected, while the infected ones can recover and become susceptible again. The implicit assumption is that there is no immunity, and recovered individuals can get infected with the same ease as those who were never infected. There are some human diseases for which this is true, for instance gonorrhea.

Transitions between the different classes of individuals can be summarized by the following scheme:



Here β is the individual rate of infection, also known as the transmission rate, and γ is the individual rate of recovery. There is an important distinction between the processes of infection and recovery: the former requires an infected individual and a susceptible individual, while the latter needs only an infected individual. Therefore, it is reasonable to suppose that the rate of growth of infected individuals is the product of the individual transmission rate β and the product of the number of infected and susceptible individuals. The overall rate of recovery is the individual recovery rate γ multiplied by the number of the infected. This leads to the following two differential equations:

$$\begin{aligned}\dot{S} &= -\beta IS + \gamma I \\ \dot{I} &= \beta IS - \gamma I\end{aligned}$$

Note that, as in the chemical kinetics models, the two equations add up to zero on the right hand

side, leading to the conclusion that $\dot{S} + \dot{I} = 0$. Therefore, the total number of people is a conserved quantity N , which does not change. This makes sense since we did not consider any births or deaths in the ODE model, only transitions between susceptible and infected individuals.

We can use the conserved quantity N to reduce the two equations to one, by the substitution of $S = N - I$:

$$\dot{I} = \beta I(N - I) - \gamma I$$

This model may be analyzed using qualitative methods, to find its fixed points and their stability and predict what fraction of the population is going to become infected for different transmission and recovery rates. We will leave this analysis to the exercises.

3.3 Analytical: Existence and uniqueness of solutions

We alluded before to the fact that differential equations are difficult to solve. In fact, mathematicians ask, can we prove a solution actually exists at all? The second question is, could there be more than one solution, satisfying a differential equation and a particular initial condition? In order for the models to make sense, we would like the ODEs to have a unique solution. There are results addressing these questions, called existence and uniqueness theorems, and we will be content to use their fruits without going through the proofs.

3.3.1 Existence of solutions

A solution for the general first-order, one variable ODE $\dot{x} = f(x, t)$ can be constructed by following the gradient field defined by the function $f(x, t)$. If the gradient field is well-behaved, there is only one, continuous trajectory to follow. By well-behaved we mean that there are no sudden jumps in direction, or no points around which there are conflicting values of $f(x, t)$. These considerations lead to the following theorem, which we state without proof:

Theorem: If $f(x, t)$ and $\frac{\partial f(x, t)}{\partial x}$ are continuous for $t \in [t_1, t_2]$ and $x \in [x_1, x_2]$, then the equation $\dot{x} = f(x, t)$ with $x(t_0) = x_0$; $t_0 \in [t_1, t_2]$; $x_0 \in [x_1, x_2]$, has a unique solution for some time interval containing t_0 .

This theorem states that as long as the defining function of an ODE is sufficiently smooth to be differentiable, a unique solution must exist at least for some time. Of course, we already know the unique solution for any linear ODE, which is valid for all time. But the existence theorem only guarantees a solution for a finite length of time. An example below illustrates a situation where a smooth, differentiable ODE only has a solution for a limited time period:

Example: Find a complete solution for the equation $\dot{x} = x^2$ with initial condition $x(0) = x_0$.

- Separate the variables:

$$\frac{dx}{x^2} = dt$$

- Integrate both sides:

$$\int \frac{dx}{x^2} = \frac{-1}{x} = \int dt = t + C$$

- Solve for the dependent variable:

$$x(t) = \frac{1}{-t - C}$$

- Using the initial condition $x(0) = x_0$, find the integration constant : $C = -1/x_0$

The complete solution is then determined, depending on the value x_0 :

$$x(t) = \frac{1}{1/x_0 - t}$$

Check for yourself that this function satisfies the ODE. Note that at time $t = 1/x_0$ there is 0 in the denominator, which means that as time approaches this value, the value of x tends toward infinity. This is known as *finite-time blow up*, and it means that the solution ceases to exist at a certain

finite time. The direction field shows that the slope of $x(t)$ becomes ever steeper as x grows, and at some point it leads the solution to approach a vertical trajectory. This leads to the question of whether and when a solution to an ODE exists.

In this example, we saw that the *interval of existence* for the solution is limited to $[0, 1/x_0)$. We have also seen examples such as $\dot{x} = x$ or $\dot{x} = x(1 - x)$, for which solutions exist for all time without blowing up. The solution to the linear equation does grow, but only at an exponential rate that does not have any finite-time singularities. The decisive difference between the two nonlinear equations $\dot{x} = x^2$ and $\dot{x} = x - x^2$ is the sign of the highest order term in the equation. When x is large, this term predominates, and if it is positive, the ever-increasing rate of growth of the dependent variable causes its value to blow up. If the highest-order term is negative, as in the logistic model, the rate of growth is negative for large x , and there can be no finite-time blow up. There may be other causes for non-existence of solutions, but it is generally a good idea to be weary of positive higher-order terms. In biological systems it is typically impossible for a quantity to explode in finite time.

Another point is that just because a solution exists in principle does not mean that it can be found analytically. Here is an example:

Example: Find the complete solution for the equation $\dot{x} = -x^3 + x^2$ with initial condition $x(0) = x_0$.

- Separate the variables:

$$\frac{dx}{-x^3 + x^2} = dt$$

- Integrate both sides:

- using partial fractions we find that

$$\frac{1}{-x^3 + x^2} = \frac{1+x}{x^2} + \frac{1}{1-x}$$

– we can integrate both sides of the equation to obtain:

$$\int \left(\frac{1+x}{x^2} + \frac{1}{1-x} \right) dx = \int \left(\frac{1}{x^2} + \frac{1}{x} + \frac{1}{1-x} \right) dx = \frac{-1}{x} + \ln|x| + \ln|1-x| = \int dt = t + C$$

- This equation cannot be solved explicitly for x .

Thus, although we have an *algebraic relation* between x and t , there is no explicit formula that can express x as a function of t . Other differential equations cannot be integrated, and in fact possess no solution that be written down as a relation. This is a typical rather than an exceptional situation for nonlinear differential equations.

3.3.2 Numerical methods and error

In practice, the most common approach to finding solutions for nonlinear differential equations is using *numerical methods*. This means using a computer program to construct a sequence of values of the dependent variable that approximate the true solution. Because a computer can only perform a finite number of operations, one cannot obtain the values of the dependent variable for a continuous range of times. Because numerical solutions are restricted to a set of discrete times, they are inherently approximate compared with the continuous time ODE.

One of the main concerns of numerical analysis is to minimize the difference between the true solution and the numerical solution, which is known as the *error*. There are at least two distinct sources of error in numerical solutions: a) *roundoff error* and b) *truncation error*.

Roundoff error is caused by the fact that real numbers are represented by a finite string of bits on a computer, using what is known as a *floating point* representation. In many programming languages variables storing real numbers can be single or double precision, which typically support 24 and 53 significant binary digits, respectively. Any arithmetic operation involving floating point numbers is only approximate, with an error that depends on the way the numbers are stored in the

memory.

Truncation error is caused by approximations inherent in numerical algorithms. For instance, numerical schemes must make approximations to convert a continuous time differential equation into a discrete time difference equation. This is frequently done by using an approximation of the derivative, such as a Newton's quotient over a short time step, to compute the next value of the dependent variable. There is a truncation error in such a scheme because it essentially uses a straight line, its slope given by the Newton's quotient, to approximate the curve of the nonlinear function in the ODE.

A numerical modeler has different controls over the roundoff error and truncation error. The first can be minimized by using more memory to store the numbers, e.g. by using double precision format for the variables. Further, there are techniques for minimizing the so-called loss of significance that occurs in certain arithmetic operations, like subtraction of two similar numbers. We will leave these considerations to numerical analysts.

Truncation error fundamentally depends on the approximation algorithm used in a numerical scheme. We can decrease the error, typically, by choosing smaller time steps, or by choosing an algorithm of higher *accuracy*. We will see the first examples of numerical schemes for ODEs in the next section. In this section, we will introduce numerical terminology for describing the pertinent qualities of numerical methods.

We would like to solve the general first order ODE:

$$\dot{x} = f(x, t); x(t_0) = x_0$$

We would like to find the solutions for $x(t)$ for a range of times $(t_0, t_1, t_2, \dots, t_N)$. Let us denote the approximate solution $x(t_i)$ by y_i . Let the time step $\Delta t = (t_N - t_0)/N$. Then a very general

expression for a discrete-time approximation of the ODE can be written as follows:

$$y_{j+1} = g(\Delta t, t_j; y_{j-m}, \dots, y_j) + \tau(\Delta t); \quad m = j, j-1, \dots, 0$$

The left hand side contains the value of y at the next time step while on the right hand side there is a function g that takes m values of y calculated at past times. This difference equation takes in the values y computed before and uses them to calculate the future value of the approximate solution y . There is one other term on the right hand side: $\tau(\Delta t)$, which is the truncation error in the scheme. The truncation error for the scheme is the difference between the value of the true solution and the one predicted by the finite difference scheme:

$$\tau(\Delta t) = |x_{j+1} - g(\Delta t, t_j; x_{j-m}, \dots, x_j)|$$

Now we can define some key terms for numerical methods:

- A numerical scheme is *consistent* if it approaches the ODE it is approximating as Δt vanishes. More precisely, the truncation error τ goes to zero:

$$\lim_{\Delta t \rightarrow 0} \tau(\Delta t) = 0$$

- A numerical scheme *converges* to the true ODE if the numerical solution $y(t)$ approaches the true solution $x(t)$ as the time step Δt vanishes:

$$\lim_{\Delta t \rightarrow 0} |x_j - y_j| = 0$$

- A numerical scheme is *stable* if the total error at the present time does not grow as the time progresses. Specifically, we can define an algorithm as stable if, for a small enough error

$$\epsilon = |x_j - y_j|,$$

$$|x_{j+1} - y_{j+1}| \leq \epsilon$$

A practical numerical scheme should have all three properties, so that it solves the correct differential equation, the error can be controlled by taking a sufficiently small time step, and the error does not grow over time. The connection between these three properties was established by a powerful result called the Lax equivalence theorem, which we present without proof:

Theorem: If a finite difference scheme is consistent with the exact ODE, and the function f is smooth and differentiable in both x and t , then the scheme is convergent if and only if it is stable.

In plain language, it states that for a reasonable finite difference scheme, convergence and stability are equivalent. Therefore, we need only worry about consistency and convergence, or consistency and stability, but not all three at the same time.

3.4 Computation: simple numerical schemes

We have introduced numerical methods in the abstract, and now we will see some specific examples. The most common class of numerical approximations is known as *finite difference* methods. As the name suggests, these methods use difference equations to approximate a differential equation. This usually involves a Taylor's series approximation of the derivative in the ODE, with variations in how many terms are used, and how the derivative is approximated.

3.4.1 Forward Euler method

All numerical methods proceed in discrete time, and some advanced methods may vary the size of the time steps. In the present examples, we will stick with a constant time step Δt . The times will be indexed by the count of the number of time steps, so $t_i = i\Delta t$. We will make a distinction between the true solution $x(t)$ and the approximate solution $y(t)$. For both letters, we will use a

subscript, x_j , y_j to denote the value of the variable at time t_j .

Here is the derivation of the simplest finite difference scheme. The first derivative of a function $x(t)$ at time step t_i can be approximated by the Newton's quotient:

$$\frac{dy_i}{dt} \approx \frac{y_{i+1} - y_i}{\Delta t}$$

This allows us to write the following *numerical scheme* to compute the numerical solution y_i of an ODE $\dot{x} = f(x, t)$:

$$y_{i+1} = y_i + \Delta t \frac{dy_i}{dt} + \epsilon(\Delta t) = y_i + \Delta t f(y_i, t_i) + \tau(\Delta t)$$

This is just a re-written Taylor's formula cut off after the first derivative, approximating the value of x at time t_{i+1} by taking the previous value and adding the derivative multiplied by the time step. Note that because we neglected the higher-order terms, we will be incurring a truncation error with every step, which we indicate by $\tau(\Delta t)$ in the equation.

This particular approximation is called the *Forward Euler Method*, after the famous mathematician who first came up with it. It is called a forward method because it uses the value of the dependent variable and its derivative at time t_i to predict the value at the next time step, t_{i+1} . This enables us to construct a solution consistent with a differential equation, given the function $f(x)$.

Pseudocode for Forward Euler

1. Define the derivative function for the ODE $f(x, t)$
2. Choose the step size Δt , and number of iterations $Niter$
3. Choose an initial point (t_0, x_0) , and initialize two arrays, T and X
4. Let $i = 1$, and repeat the following while $i < Niter$:
 - (a) store $X[i] + \Delta t * f(X[i], T[i])$ in $X[i + 1]$

(b) store $T[i] + \Delta t$ in $T[i + 1]$

(c) increase i by 1

Example: To numerically solve the equation $\dot{x} = ax$, we substitute the function ax for the function $f(x, t)$, and obtain a scheme approximating a particular ODE:

$$y_{i+1} = y_i + \Delta t a y_i = (1 + a\Delta t)y_i$$

We had introduced the concepts of consistency, convergence and stability. We used a standard Taylor's approximation for the derivative, so consistency is assured as the time step vanishes. Now we need to test either for stability or convergence, since we had shown that they are equivalent for a consistent scheme.

Let us address the question of stability for this scheme. If the numerical solution at time t_i has error ϵ , we can write it as follows: $y_i = x_i + \epsilon$. Then at the next time step the value of the solution will be:

$$y_{i+1} = (x_i + \epsilon)(1 + a\Delta t) = x_i(1 + a\Delta t) + \epsilon(1 + a\Delta t)$$

If we assume $a > 0$, then the additive error is multiplied by a number greater than 1, and will therefore grow. Of course, the truncation and machine error from this step may reduce the total error sometimes, but this is not something to rely on. In fact, as the scheme is propagated through time, the error keeps growing exponentially, as in the solution of the linear difference equation in Chapter 1. Forward Euler is in fact unstable for any value of Δt in this example.

Example: Let us numerically solve $\dot{x} = -ax$, by substituting $-ax$ into the function $f(x, t)$ in Forward Euler. We obtain the following scheme:

$$y_{i+1} = y_i - \Delta t a y_i = (1 - a\Delta t)y_i$$

Again, let us investigate the stability of this scheme:

$$y_{i+1} = (x_i + \epsilon)(1 - a\Delta t) = x_i(1 - a\Delta t) + \epsilon(1 - a\Delta t)$$

Now the multiplicative constant on the right is always less than 1, but depending on the value of Δt , it may be greater than 1 in absolute value. Set up the following inequality to assure the multiplicative constant does not increase the error:

$$|1 - a\Delta t| < 1 \Rightarrow \Delta t < 2/a$$

Thus, the Forward Euler method in this example is stable if the time step Δt is sufficiently small. Generally speaking, however, Forward Euler is about the worst method to use for practical numerical solutions of ODEs.

3.4.2 Backward Euler method

More sophisticated numerical methods generally offer better stability than Forward Euler. For instance, there is a class of methods called *implicit* schemes which rely on evaluating the value of the derivative of x at a future time point. This may seem impossible, since we do not yet have the value of the dependent variable x in the future, only in the present. In fact, we can set up an algebraic relationship between the present value of x , the future value of x , and the derivative of x in the future. Then, depending on the form of the defining function $f(x)$, we may solve this relationship for the value of x at the future time.

To make the idea of implicit methods concrete, we will introduce a simple method called the Backward Euler. As suggested by the name, this method is essentially similar to the Forward Euler,

but with the future value of x_{i+1} substituted in the defining function instead of the current value:

$$y_{i+1} = y_i + \Delta t \frac{dy_{i+1}}{dt} = y_i + \Delta t f(y_{i+1})$$

How can we calculate the value of $f(y_{i+1})$ without knowing y_{i+1} ? Depending on the form of $f(y)$, it is sometimes possible to algebraically solve for y_{i+1} . If we can solve the implicit expression for y_{i+1} , we can program a numerical scheme that will compute the value y_{i+1} directly from y_i . In other situations, the implicit expression may be impossible to solve algebraically. The practitioner may then use a method for solving such an expression numerically, such as Newton's method that we will see in Unit 4.

Pseudocode for Backward Euler

1. Define the derivative function for the ODE $f(x, t)$
2. Choose the step size Δt , and number of iterations $Niter$
3. Choose an initial point (t_0, x_0) , and initialize two arrays, T and X
4. Let $i = 1$, and repeat the following while $i < Niter$:
 - (a) solve $X[i] + \Delta t * f(X[i + 1]) = X[i + 1]$ for $X[i + 1]$ and store the value
 - (b) store $T[i] + \Delta t$ in $T[i + 1]$
 - (c) increase i by 1

Example: Here is the implementation of the Backward Euler for the linear ODE $\dot{x} = ax$:

$$y_{i+1} = y_i + \Delta t a y_{i+1}$$

For this particular situation, the implicit equation can be solved for the future value y_{i+1} :

$$(1 - a\Delta t)y_{i+1} = y_i \implies y_{i+1} = \frac{1}{1 - a\Delta t}y_i$$

Now we use the same stability analysis as we did for Forward Euler: assume the numerical solution y_i has total error ϵ , and substitute $y_i = x_i + \epsilon$:

$$y_{i+1} = \frac{1}{1 - a\Delta t}(x_i + \epsilon) = \frac{1}{1 - a\Delta t}x_i + \epsilon \frac{1}{1 - a\Delta t}$$

Assume that $a > 0$, the Backward Euler scheme for the exponential growth ODE is stable when Δt : $|1 - a\Delta t| > 1 \implies \Delta t > 2/a$. This appears counterintuitive, because the scheme is unstable for small time steps.

Example: We also apply Backward Euler scheme to $\dot{x} = -ax$, and again solve for the future value y_{i+1} :

$$y_{i+1} = y_i - \Delta t a y_{i+1} \implies y_{i+1} = \frac{1}{1 + a\Delta t}y_i$$

Substitute the solution y_i with error ϵ , we find that the error propagates as follows:

$$y_{i+1} = \frac{1}{1 + a\Delta t}(x_i + \epsilon) = \frac{1}{1 + a\Delta t}x_i + \epsilon \frac{1}{1 + a\Delta t}$$

3.5 Synthesis: Lacker model of ovarian follicle maturation

We will now consider mathematical modeling used to understand physiology. As our example we will take Lacker's model of hormonal control of the maturation and release of eggs in mammalian ovaries. Ovarian follicles are the basic units of the ovaries. Each contains a single oocyte, which is released during ovulation after it matures. Ovaries contain numerous follicles, with only a few, and in humans usually only one, maturing during each reproductive cycle. In fact the vast majority of

follicles never mature at all, but instead atrophy and die. What is the mechanism for this selectivity in maturation?

The physiological control of this process is achieved through blood-borne hormones, secreted by the pituitary gland in the brain, and by the ovarian follicles themselves. The pituitary releases gonadotropin, which consists of two different hormones, follicle-stimulating hormone (FSH), and luteinizing hormone (LH). Gonadotropin stimulates the maturation of follicles, which in turn secrete the hormone estradiol. This hormone stimulates the production of gonadotropin, causing increased secretion of estradiol, etc. This self-reinforcing control mechanism is called a *positive feedback loop*.

As follicles mature, they become more sensitive to gonadotropin, and their rate of estradiol secretion increases concomitantly. We will consider the production of estradiol by each follicle as a proxy variable for follicle maturity. Let $x(t)$ be the total concentration of estradiol in the bloodstream, and let $x_i(t)$ be the contribution of the i -th follicle to the total concentration. If there are N total follicles, then the total concentration is the sum of the contributions from each of N follicles:

$$x(t) = \sum_{i=1}^N x_i(t)$$

Let us further suppose that each follicle communicates with the rest only through the bloodstream, and therefore the rate of secretion of estradiol depends on the production in the follicle itself (x_i) and on the total concentration in the blood x . In other words, for each follicles we can write a differential equation with a function that depends on those two variables: $\dot{x}_i = f(x, x_i)$. It turns out that there is a particular form of this function that matches experimental data, and it looks like this:

$$\dot{x}_i = Cx_i[1 - D(x - M_1x_i)(x - M_2x_i)]$$

The parameters C , D , M_1 , and M_2 are the same for each follicle in this simple model. The main question to understand is how only a few follicles mature, despite all having the same physiological

properties.

There are as many differential equations in the model as there are follicles, and as many dependent variables x_i . Let us first analyze the model in the simple case of a single follicle. Then $x_1 = x$ is the total concentration of estradiol, and the ODE is:

$$\dot{x} = Cx[1 - D(x - M_1x)(x - M_2x)] = Cx - D(1 - M_1)(1 - M_2)x^3$$

This is a nonlinear differential equation, with the highest term of the cubic power. We can analyze its behavior qualitatively by finding the fixed points and their stability. The fixed points are solutions to the following algebraic equation:

$$Cx - D(1 - M_1)(1 - M_2)x^3 = 0$$

The first fixed point is at $x = 0$, which corresponds to a dead follicle not producing any hormones. The others are solutions to the quadratic equation $1 = D(1 - M_1)(1 - M_2)x^2$. The solutions are $x = \pm\sqrt{1/(D(1 - M_1)(1 - M_2))}$. The negative solution of course has no biological meaning, so we are left with a single non-zero fixed point for estradiol production.

We can analyze the stability of each fixed point by finding the derivative of the defining function at each point:

$$f'(x) = C - 3D(1 - M_1)(1 - M_2)x^2$$

Substituting the fixed point $x = 0$ we find that $f'(0) = C$, which means the fixed point is unstable if $C > 0$. For the other fixed point $x = \sqrt{1/(D(1 - M_1)(1 - M_2))}$, the value of the derivative is: $f'(\sqrt{1/(D(1 - M_1)(1 - M_2))}) = C - 3$. Thus, for $0 < C < 3$, the nonzero fixed point is stable.

The preceding exercise has little to do with actual model, since the whole point is to understand selective maturation of a few follicles out of many. In order to investigate this phenomenon, we will use numerical methods such as Backward Euler to predict the maturation dynamics of different

follicles.