1
Difference equations

The world is too rich and complex for our minds to grasp it whole, for our minds are but a small part of the richness of the world. To cope with the complexity, we reason hierarchically. We divide the world into small, comprehensible pieces: systems. Systems are ubiquitous: a CPU, a memory chips, a motor, a web server, a jumbo jet, the solar system, the telephone system, or a circulatory system. Systems are a useful abstraction, chosen because their external interactions are weaker than their internal interactions. That properties makes independent analysis meaningful.

Systems interact with other systems via forces, messages, or in general via information or signals. 'Signals and systems' is the study of systems and their interaction.

This book studies only discrete-time systems, where time jumps rather than changes continuously. This restriction is not as severe as its seems. First, digital computers are, by design, discrete-time devices, so discrete-time signals and systems includes digital computers. Second, almost all the important ideas in discrete-time systems apply equally to continuous-time systems.

Alas, even discrete-time systems are too diverse for one method of analysis. Therefore even the abstraction of systems needs subdivision. The particular class of so-called linear and time-invariant systems admits powerful tools of analysis and design. The benefit of restricting ourselves to such

systems, and the meaning of the restrictions, will become clear in subsequent chapters.

1.1 Rabbits

Here is Fibonacci's problem [6, 10], a famous discrete-time, linear, time-invariant system and signal:

> A certain man put a pair of rabbits in a place surrounded on all sides by a wall. How many pairs of rabbits can be produced from that pair in a year if it is supposed that every month each pair begets a new pair which from the second month on becomes productive?

1.1.1 Mathematical representation

This system consists of the rabbit pairs and the rules of rabbit reproduction. The signal is the sequence f where $f[n]$ is the number of rabbit pairs at month n (the problem asks about $n = 12$).

▶ *What is f in the first few months?*

In month 0, one rabbit pair immigrates into the system: $f[0] = 1$. Let's assume that the immigrants are children. Then they cannot have their own children in month 1 – they are too young – so $f[1] = 1$. But this pair is an adult pair, so in month 2 the pair has children, making $f[2] = 2$.

Finding $f[3]$ requires considering the adult and child pairs separately (hierarchical reasoning), because each type behaves according to its own reproduction rule. The child pair from month 2 grows into adulthood in month 3, and the adult pair from month 2 begets a child pair. So in $f[3] = 3$: two adult and one child pair.

The two adult pairs contribute two child pairs in month 4, and the one child pair grows up, contributing an adult pair. So month 4 has five pairs: two child and three adult pairs. To formalize this reasoning process, define two intermediate signals c and a:

$c[n] =$ number of child pairs at month n;
$a[n] =$ number of adult pairs at month n.

The total number of pairs at month n is $f[n] = c[n] + a[n]$. Here is a table showing the three signals c, a, and f:

n	0	1	2	3
c	1	0	1	1
a	0	1	1	2
f	1	1	2	3

The arrows in the table show how new entries are constructed. An upward diagonal arrow represents the only means to make new children, namely from last month's adults:

$$a[n-1] \to c[n] \quad \text{or} \quad c[n] = a[n-1].$$

A horizontal arrow represents one contribution to this month's adults, that adults last month remain adults: $a[n-1] \to a[n]$. A downward diagonal arrow represents the other contribution to this month's adults, that last month's children grow up into adults: $c[n-1] \to a[n]$. The sum of the two contributions is

$$a[n] = a[n-1] + c[n-1].$$

▶ *What is the difference equation for f itself?*

To find the equation for f, one has at least two methods: logical deduction (Problem 1.1) or trial and error. Trial and error is better suited for finding results, and logical deduction is better suited for verifying them. Therefore, using trial and error, look for a pattern among addition samples of f:

n	0	1	2	3	4	5	6
c	1	0	1	1	2	3	5
a	0	1	1	2	3	5	8
f	1	1	2	3	5	8	13

▶ *What useful patterns live in these data?*

One prominent pattern is that the signals c, a, and f look like shifted versions of each other:

$$a[n] = f[n-1];$$
$$c[n] = a[n-1] = f[n-2].$$

Since $f[n] = a[n] + c[n]$,

$$f[n] = f[n-1] + f[n-2].$$

with initial conditions $f[0] = f[1] = 1$.

This mathematical description, or representation, clarifies a point that is not obvious in the verbal description: that the number of rabbit pairs in any month depends on the number in the two preceding months. This difference equation is said to be a second-order difference equation. Since its coefficients are all unity, and the signs are positive, it is the simplest second-order difference equation. Yet its behavior is rich and complex.

Problem 1.1   Verifying the conjecture
Use the two intermediate equations

$$c[n] = a[n-1],$$
$$a[n] = a[n-1] + c[n-1];$$

and the definition $f[n] = a[n] + c[n]$ to confirm the conjecture

$$f[n] = f[n-1] + f[n-2].$$


1.1.2   Closed-form solution

The rabbit difference equation is an implicit recipe that computes new values from old values. But does it have a **closed form**: an explicit formula for $f[n]$ that depends on n but not on preceding samples? As a step toward finding a closed form, let's investigate how $f[n]$ behaves as n becomes large.

▶ *Does $f[n]$ grow like a polynomial in n, like a logarithmic function of n, or like an exponential function of n?*

Deciding among these options requires more data. Here are many values of $f[n]$ (starting with month 0):

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, \ldots$$

The samples grow quickly. Their growth is too rapid to be logarithmic, unless $f[n]$ is an unusual function like $(\log n)^{20}$. Their growth is probably also too rapid for $f[n]$ to be a polynomial in n, unless $f[n]$ is a high-degree polynomial. A likely alternative is exponential growth. To test that hypothesis, use pictorial reasoning by plotting $\ln f[n]$ versus n. The plotted points oscillate above and below a best-fit straight line. Therefore $\ln f[n]$ grows almost exactly linearly with n, and $f[n]$ is approximately an exponential function of n:
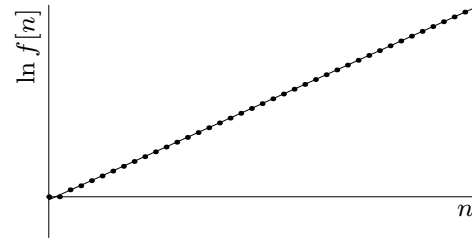
$$f[n] \approx Az^n,$$

where z and A are constants.

▶ *How can z be estimated from $f[n]$ data?*

Because the plotted points fall ever closer to the best-fit line as n grows, the exponential approximation $f[n] \approx Az^n$ becomes more exact as n grows. If the approximation were exact, then $f[n]/f[n-1]$ would always equal z, so $f[n]/f[n-1]$ becomes an ever closer approximation to z as n increases.

n	$f[n]/f[n-1]$
10	1.6181818181818
20	1.6180339985218
30	1.6180339887505
40	1.6180339887499
50	1.6180339887499

These ratios seem to converge to $z = 1.6180339887499$.

Its first few digits 1.618 might be familiar. For a memory amplifier, feed the ratio to the online Inverse Symbolic Calculator (ISC). Given a number, it guesses its mathematical source. When given the Fibonacci z, the Inverse Symbolic Calculator suggests two equivalent forms: that z is a root of $1 - x - x^2$ or that it is $\phi \equiv (1 + \sqrt{5})/2$. The constant ϕ is the famous golden ratio [5].

Therefore, $f[n] \approx A\phi^n$. To find the constant of proportionality A, take out the big part by dividing $f[n]$ by ϕ^n. These ratios hover around $0.723\ldots$, so perhaps A is $\sqrt{3} - 1$. Alas, experiments with larger values of n strongly suggest that the digits continue $0.723606\ldots$ whereas $\sqrt{3} - 1 = 0.73205\ldots$. A bit of experimentation or the Inverse Symbolic Calculator suggests that 0.72360679774998 probably originates from $\phi/\sqrt{5}$.

n	$f[n]/f[n-1]$
10	0.72362506926472
20	0.72360679895785
30	0.72360679775006
40	0.72360679774998
50	0.72360679774998

This conjecture has the merit of reusing the $\sqrt{5}$ already contained in the definition of ϕ, so it does not introduce a new arbitrary number. With that conjecture for A, the approximation for $f[n]$ becomes

$$f[n] \approx \frac{\phi^{n+1}}{\sqrt{5}}.$$

▶ *How accurate is this approximation?*

To test the approximation, take out the big part by computing the residual:

$$r[n] = f[n] - \phi^{n+1}/\sqrt{5}.$$

The residual decays rapidly, perhaps exponentially. Then r has the general form

$$r[n] \approx By^n,$$

where y and B are constants. To find y, compute the ratios $r[n]/r[n-1]$. They converge to $-0.61803\ldots$, which is almost exactly $1 - \phi$ or $-1/\phi$. Therefore $r[n] \approx B(-1/\phi)^n$.

n	$r[n]/r[n-1]$
2	-0.61803398874989601
3	-0.61803398874989812
4	-0.61803398874988624
5	-0.61803398874993953
6	-0.61803398874974236
7	-0.61803398875029414
8	-0.61803398874847626
9	-0.61803398875421256
10	-0.61803398873859083

▶ *What is the constant of proportionality B?*

To compute B, divide $r[n]$ by $(-1/\phi)^n$. These values, if n is not too large (Problem 1.2), almost instantly settles on 0.27639320225. With luck, this number can be explained using ϕ and $\sqrt{5}$. A few numerical experiments suggest the conjecture

$$B = \frac{1}{\sqrt{5}} \times \frac{1}{\phi}.$$

The residual becomes

$$r[n] = -\frac{1}{\sqrt{5}} \times \left(-\frac{1}{\phi}\right)^{n+1}.$$

The number of rabbit pairs is the sum of the approximation Az^n and the residual By^n:

$$f[n] = \frac{1}{\sqrt{5}} \left(\phi^{n+1} - (-\phi)^{-(n+1)}\right).$$

How bizarre! The Fibonacci signal f splits into two signals in at least two ways. First, it is the sum of the adult-pairs signal a and the child-pairs signal c. Second, it is the sum $f_1 + f_2$ where f_1 and f_2 are defined by

$$f_1[n] \equiv \frac{1}{\sqrt{5}} \phi^{n+1};$$

$$f_2[n] \equiv -\frac{1}{\sqrt{5}} (-1/\phi)^{n+1}.$$

The equivalence of these decompositions would have been difficult to predict. Instead, many experiments and guesses were needed to establish the equivalence. Another kind of question, also hard to answer, arises by changing merely the plus sign in the Fibonacci difference equation into a minus sign:

$$g[n] = g[n-1] - g[n-2].$$

With corresponding initial conditions, namely $g[0] = g[1] = 1$, the signal g runs as follows (staring with $n = 0$):

$$\underbrace{1, 1, 0, -1, -1, 0, 1}_{\text{one period}}, 1, 0, -1, -1, 0, \dots.$$

Rather than growing approximately exponentially, this sequence is exactly periodic. Why? Furthermore, it has period 6. Why? How can this period be predicted without simulation?

A representation suited for such questions is introduced in ??. For now, let's continue investigating difference equations to represent systems.

> **Problem 1.2   Actual residual**
>
> Here is a semilog graph of the absolute residual $|r[n]|$ computed numerically up to $n = 80$. (The absolute residual is used because the residual is often negative and would have a complex logarithm.) It follows the predicted exponential decay for a while, but then misbehaves. Why?
>
>
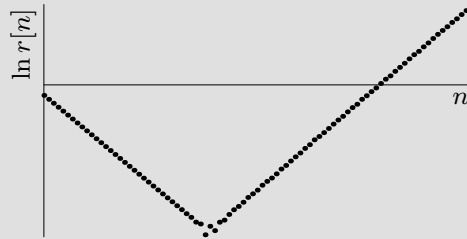
1.2 Leaky tank

In the Fibonacci system, the rabbits changed their behavior – grew up or had children – only once a month. The Fibonacci system is a discrete-time

system. These systems are directly suitable for computational simulation and analysis because digital computers themselves act like discrete-time systems. However, many systems in the world – such as piano strings, earthquakes, microphones, or eardrums – are naturally described as continuous-time systems.

To analyze continuous-time systems using discrete-time tools requires approximations. These approximations are illustrated in the simplest interesting continuous-time system: a leaky tank. Imagine a bathtub or sink filled to a height h with water. At time $t = 0$, the drain is opened and water flows out. What is the subsequent height of the water?
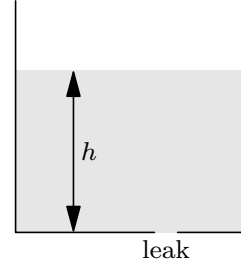
At $t = 0$, the water level and therefore the pressure is at its highest, so the water drains most rapidly at $t = 0$. As the water drains and the level falls, the pressure and the rate of drainage also fall. This behavior is captured by the following differential equation:

$$\frac{dh}{dt} = -f(h),$$

where $f(h)$ is an as-yet-unknown function of the height.

Finding $f(h)$ requires knowing the geometry of the tub and piping and then calculating the flow resistance in the drain and piping. The simplest model for resistance is a so-called linear leak: that $f(h)$ is proportional to h. Then the differential equation simplifies to

$$\frac{dh}{dt} \propto -h.$$

▶ *What are the dimensions of the missing constant of proportionality?*

The derivative on the left side has dimensions of speed (height per time), so the missing constant has dimensions of inverse time. Call the constant $1/\tau$, where τ is the **time constant** of the system. Then

$$\frac{dh}{dt} = -\frac{h}{\tau}.$$

An almost-identical differential equation describes the voltage V on a capacitor discharging across a resistor:

$$\frac{dV}{dt} = -\frac{1}{RC} V.$$
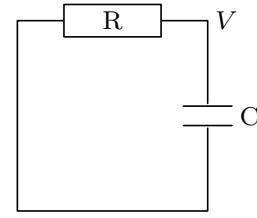
It is the leaky-tank differential equation with time constant $\tau = RC$.

> **Problem 1.3   Kirchoff's laws**
> Use Kirchoff's laws to verify this differential equation.

Approximating the continuous-time differential equation as a discrete-time system enables the system to be simulated by hand and computer. In a discrete-time system, time advances in lumps.

If the lump size, also known as the timestep, is T, then h[n] is the discrete-time approximation of h(nT). Imagine that the system starts with $h[0] = h_0$. What is h[1]? In other words, what is the discrete-time approximation for h(T)? The leaky-tank equation says that

$$\frac{dh}{dt} = -\frac{h}{\tau}.$$

At $t = 0$ this derivative is $-h_0/\tau$. If dh/dt stays fixed for a whole timestep – a slightly dubious but simple assumption – then the height falls by approximately $h_0 T/\tau$ in one timestep. Therefore

$$h[1] = h_0 - \frac{T}{\tau} h_0 = \left(1 - \frac{T}{\tau}\right) h[0].$$

▶ *Using the same assumptions, what is h[2] and, in general, h[n]?*

The reasoning to compute h[1] from h[0] applies when computing h[2] from h[1]. The derivative at $n = 1$ – equivalently, at $t = T$ – is $-h[1]/\tau$. Therefore between $n = 1$ and $n = 2$ – equivalently, between $t = T$ and $t = 2T$ – the height falls by approximately $-h[1]T/\text{tau}$,

$$h[2] = h_1 - \frac{T}{\tau} h_1 = \left(1 - \frac{T}{\tau}\right) h[1].$$

This pattern generalizes to a rule for finding every h[n]:

$$h[n] = \left(1 - \frac{T}{\tau}\right) h[n-1].$$

This implicit equation has the closed-form solution

$$h[n] = h_0 \left(1 - \frac{T}{\tau}\right)^n.$$

▶ *How closely does this solution reproduce the behavior of the original, continuous-time system?*

The original, continuous-time differential equation $dh/dt = -h\tau u$ is solved by

$$h(t) = h_0 e^{-t/\tau}.$$

At the discrete times $t = nT$, this solution becomes

$$h(t) = h_0 e^{-nT/\tau} = h_0 \left(e^{-T/\tau}\right)^n.$$

The discrete-time approximation replaces $e^{-T/\tau}$ with $1 - T/\tau$. That difference is the first two terms in the Taylor series for $e^{-T/\tau}$:

$$e^{-T/\tau} = 1 - \frac{T}{\tau} + \frac{1}{2}\left(\frac{T}{\tau}\right)^2 - \frac{1}{6}\left(\frac{T}{\tau}\right)^3 + \dots.$$

Therefore the discrete-time approximation is accurate when the higher-order terms in the Taylor series are small – namely, when $T/\tau \ll 1$.

This method of solving differential equations by replacing them with discrete-time analogs is known as the Euler approximation, and it can be used to solve equations that are very difficult or even impossible to solve analytically.

Problem 1.4   Which is the approximate solution?
Here are unlabeled graphs showing the discrete-time samples $h[n]$ and the continuous-time samples $h(nT)$, for $n = 0 \dots 6$. Which graph shows the discrete-time signal?
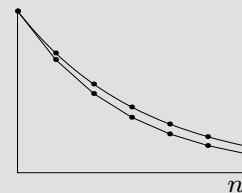
Problem 1.5   Large timesteps
Sketch the discrete-time samples $h[n]$ in three cases: (a.) $T = \tau/2$ (b.) $T = \tau$ (c.) $T = 2\tau$ (d.) $T = 3\tau$

Problem 1.6   Tiny timesteps
Show that as $T \to 0$, the discrete-time solution

$$h[n] = h_0 \left(1 - \frac{T}{\tau}\right)^n.$$

approaches the continuous-time solution

$$h(t) = h_0 e^{-nT/\tau}.$$

How small does T have to be, as a function of n, in order that the two solutions approximately match?

1.3 Fall of a fog droplet

The leaky tank (Section 1.2) is a first-order system, and its differential equation and difference equation are first-order equations. However, the physical world is often second order because Newton's second law of motion, $F = ma$, contains a second derivative.

For such systems, how applicable is the Euler approximation? To illustrate the issues that arise in applying the Euler approximation to second-order systems, let's simulate the fall of a fog droplet acted on by gravity ($F = mg$) and air resistance. A fog droplet is small enough that its air resistance is proportional to velocity rather than to the more usual velocity squared. Then the net downward force on the droplet is $mg - bv$, where v is its velocity and b is a constant that measures the strength of the drag. In terms of position x, with the positive direction as downward, Newton's second law becomes

$$m\frac{d^2x}{dt^2} = mg - b\frac{dx}{dt}.$$

Dividing both sides by m gives

$$\frac{d^2x}{dt^2} = g - \frac{b}{m}\frac{dx}{dt}.$$

▶ *What are the dimensions of b/m?*

The constant b/m turns the velocity dx/dt into an acceleration, so b/m has dimensions of inverse time. Therefore rewrite it as $1/\tau$, where $\tau \equiv m/b$ is a time constant. Then

$$\frac{d^2x}{dt^2} = g - \frac{1}{\tau}\frac{dx}{dt}.$$

▶ *What is a discrete-time approximation for the second derivative?*

In the leaky-tank equation,

$$\frac{dh}{dt} = -\frac{h}{\tau},$$

the first derivative at $t = nT$ had the Euler approximation $(h[n+1] - h[n])/T$ and $h(t = nT)$ became $h[n]$. The second derivative d^2x/dt^2 is the limit of a difference of two first derivatives. Using the Euler approximation procedure, approximate the first derivatives at $t = nT$ and $t = (n+1)T$:

$$\left.\frac{dx}{dt}\right|_{t=nT} \approx \frac{x[n+1] - x[n]}{T};$$
$$\left.\frac{dx}{dt}\right|_{t=(n+1)T} \approx \frac{x[n+2] - x[n+1]}{T}.$$

Then

$$\left.\frac{d^2x}{dt^2}\right|_{t=nT} \approx \frac{1}{T}\left(\frac{x[n+2] - x[n+1]}{T} - \frac{x[n+1] - x[n]}{T}\right).$$

This approximation simplifies to

$$\left.\frac{d^2x}{dt^2}\right|_{t=nT} \approx \frac{1}{T^2}\left(x[n+2] - 2x[n+1] + x[n]\right).$$

The Euler approximation for the continuous-time equation of motion is then

$$\frac{1}{T^2}\left(x[n+2] - 2x[n+1] + x[n]\right) = g - \frac{1}{\tau}\left(\frac{x[n+1] - x[n]}{T}\right)$$

or

$$x[n+2] - 2x[n+1] + x[n] = gT^2 - \frac{T}{\tau}(x[n+1] - x[n]).$$

Our old friend from the leaky tank, the ratio T/τ, has reappeared in this problem. To simplify the subsequent equations, define $\alpha \equiv T/\tau$. Then after collecting the like terms, the difference equation for the falling fog droplet is
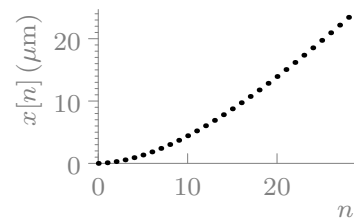
$$x[n+2] = (2 - \alpha)x[n+1] - (1 - \alpha)x[n] + gT^2.$$

As expected, this difference equation is second order. Like the previous second-order equation, the Fibonacci equation, it needs two initial values. Let's try $x[-1] = x[0] = 0$. Physically, the fog droplet starts from rest at the reference height 0, and at $t = 0$ starts feeling the gravitational force mg.

For a typical fog droplet with radius $10\,\mu m$, the physical parameters are:

$$m \sim 4.2 \cdot 10^{-12}\,\text{kg};$$
$$b \sim 2.8 \cdot 10^{-9}\,\text{kg}\,\text{s}^{-1};$$
$$\tau \sim 1.5 \cdot 10^{-3}\,\text{s}^{-1}.$$

Relative to τ, the timestep T should be small, otherwise the simulation error will large. A timestep such as $0.1\,\text{ms}$ is a reasonable compromise between keeping reducing the error and speeding up the simulation. Then the dimensionless ratio α is 0.0675. A simulation using these parameters shows x initially rising faster than linearly, probably quadratically, then rising linearly at a rate of roughly $1.5\,\mu m$ per timestep or $1.5\,\text{cm}\,\text{s}^{-1}$.

This simulation result explains the longevity of fog. Fog is, roughly speaking, a cloud that has sunk to the ground. Imagine that this cloud reaches up to $500\,\text{m}$ (a typical cloud thickness). Then, to settle to the ground, the cloud requires

$$t_{\text{fall}} \sim \frac{500\,\text{m}}{1.5\,\text{cm}\,\text{s}^{-1}} \sim 9\,\text{hours}.$$

In other words, fog should last overnight – in agreement with experience!

Counting timesteps How many timesteps would the fog-droplet simulation require (with $T = 0.1\,\text{ms}$) in order for the droplet to fall $500\,\text{m}$ in the simulation? How long would your computer, or another easily available computer, require to simulate that many timesteps?

Problem 1.7 Terminal velocity

By simulating the fog equation

$$x[n + 2] = (2 - \alpha)x[n + 1] - (1 - \alpha)x[n] + gT^2.$$

with several values of T and therefore α, guess a relation between g, T, α, and the terminal velocity of the particle.

1.4 Springs

Now let's extend our simulations to the most important second-order system: the spring. Springs are a model for a vast number of systems in the natural and engineered worlds: planetary orbits, chemical bonds, solids, electromagnetic radiation, and even electron–proton bonds. Since color results from electromagnetic radiation meeting electron–proton bonds, grass is green and the sky is blue because of how springs interact with springs.

The simplest spring system is a mass connected to a spring and free to oscillate in just one dimension. Its differential equation is
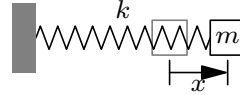
$$m\frac{d^2x}{dt^2} + kx = 0,$$

where x is the block's displacement from the equilibrium position, m is the block's mass, and k is the spring constant. Dividing by m gives

$$\frac{d^2x}{dt^2} + \frac{k}{m}x = 0.$$

Defining the angular frequency $\omega \equiv \sqrt{k/m}$ gives the clean equation:

$$\frac{d^2x}{dt^2} + \omega^2 x = 0.$$

Now divide time into uniform steps of duration T, and replace the second derivative d^2x/dt^2 with a discrete-time approximation:

$$\left.\frac{d^2x}{dt^2}\right|_{t=nT} \approx \frac{x[n+2] - 2x[n+1] + x[n]}{T^2},$$

where as usual the sample $x[n]$ corresponds to the continuous-time signal $x(t)$ at $t = nT$. Then

$$\frac{x[n+2] - 2x[n+1] + x[n]}{T^2} + \omega^2 x[n] = 0$$

or after collecting like terms,

$$x[n+2] = 2x[n+1] - \left(1 + (\omega T)^2\right)x[n].$$

Defining $\alpha \equiv \omega T$,

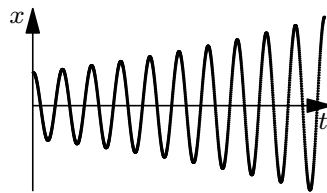$$x[n+2] = 2x[n+1] - \left(1 + \alpha^2\right)x[n].$$

This second-order difference equation needs two initial values. A simple pair is $x[0] = x[1] = x_0$. This choice corresponds to pulling the mass rightwards by x_0, then releasing it at $t = T$. What happens afterward?

To simulate the system numerically, one should choose T to make α small. As a reasonable small α, try 100 samples per oscillation period: $\alpha = 2\pi/100$ or roughly 0.06. Alas, the simulation predicts that the oscillations grow to infinity.
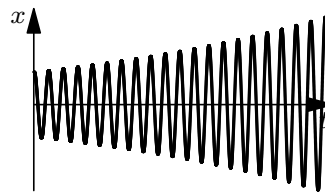
```
name: dummy

file: shm-forward

state: unknown
```

▶ *What went wrong?*

Perhaps α, even at 0.06, is too large. Here are two simulations with smaller values of α:

$$\alpha \approx 0.031 \qquad\qquad \alpha \approx 0.016$$

These oscillations also explode. The only difference seems to be the rate of growth (Problem 1.8).

Problem 1.8   Tiny values of α

Simulate

$$x[n + 2] = 2x[n + 1] - (1 + \alpha^2)x[n]$$

using very small values for α. What happens?

An alternative explanation is that the discrete-time approximation of the derivative caused the problem. If so, it would be surprising, because the same approximation worked when simulating the fall of a fog droplet. But let's try an alternative definition: Instead of defining

$$\left.\frac{dx}{dt}\right|_{t=nT} \approx \frac{x[n + 1] - x[n]}{T},$$

try the simple change to

$$\frac{dx}{dt} \approx \frac{x[n] - x[n - 1]}{T}.$$

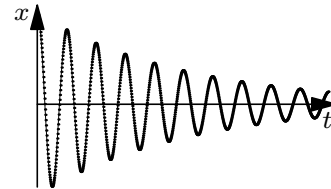Using the same procedure for the second derivative,

$$\frac{d^2x}{dt^2}\bigg|_{t=nT} \approx \frac{x[n] - 2x[n-1] + x[n-2]}{T^2}.$$

The discrete-time spring equation is then
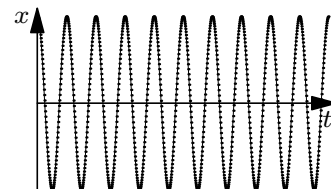
$$(1 + \alpha^2)x[n] = 2x[n-1] - x[n-2],$$

or

$$x[n] = \frac{2x[n-1] - x[n-2]}{1 + \alpha^2}.$$

Using the same initial conditions $x[0] = x[1] = 1$, what is the subsequent time course? The bad news is that these oscillations decay to zero!

However, the good news is that changing the derivative approximation can significantly affect the behavior of the discrete-time system. Let's try a symmetric second derivative:

$$\frac{d^2x}{dt^2}\bigg|_{t=nT} \approx \frac{x[n+1] - 2x[n] + x[n-1]}{T^2}.$$

Then the difference equation becomes

$$x[n+2] = (2 - \alpha^2)x[n+1] - x[n].$$

Now the system oscillates stably, just as a spring without energy loss or input should behave.

Why did the simple change to a symmetric second derivative solve the problem of decaying or growing oscillations? The representation of the alternative discrete-time systems as difference equations does not help answer that question. Its answer requires the two most important ideas in signals and systems: operators (??) and modes (??).

Problem 1.9   Different initial conditions

Here are the subsequent samples using the symmetric second derivative and initial conditions $x[0] = 0$, $x[1] = x_0$. The amplitude is, however, much larger than x_0. Is that behavior physically reasonable? If yes, explain why. If not, explain what should happen.
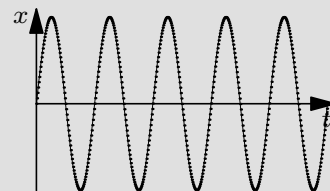
MIT OpenCourseWare

6.003 Signals and Systems
Spring 2010