Analysis of Discrete Models Using an Electronic Spreadsheet—A Survey of Examples

Thomas C. McMillan
tcmcmillan@ualr.edu
Department of Mathematics & Statistics
University of Arkansas at Little Rock
72204-1099
U.S.A.

Abstract

This paper surveys examples in which questions are answered by developing recurrence relations that model certain situations and then exploring those recurrence relations using an electronic spreadsheet. The goals of this approach are (1) to develop an understanding of and a facility for using recurrence relations as a modeling tool to motivate the need to derive analytical methods for solving recurrence relations, (2) to increase awareness of the spreadsheet as a problem solving tool, (3) to validate intermediate steps in analytical solutions using the spreadsheet, and (4) to use the spreadsheet's computational and graphics capabilities to explore and analyze questions that do not lend themselves to an easy analytical solution. The examples described here have been used in discrete mathematics courses and professional development courses for licensed teachers.

1 Introduction

In this paper we discuss and illustrate through examples how to use an electronic spreadsheet to analyze discrete models. The ease with which formulas can be entered into the spreadsheet's data array make it an ideal tool for exploring recurrence relations with one or two independent variables. Students in discrete mathematics courses can, using the spreadsheet, numerically solve recurrence relations and explore their properties with little programming effort. The advantages of incorporating the spreadsheet into a discrete mathematics courses are many. Perhaps most important is that students can solve recurrence relations numerically before studying the analytical techniques. Thus students can see early on real applications of discrete mathematics, explore the resulting models, and, in the process, develop a motivation for learning the analytic techniques. Representing the solutions to discrete models in a graph is relatively easy. When analytical methods are developed, the spreadsheet models can be adapted to illustrate the required steps. Finally, students can develop an appreciation for the spreadsheet as a powerful problem solving tool by applying it to problems that are difficult or impossible to solve analytically.

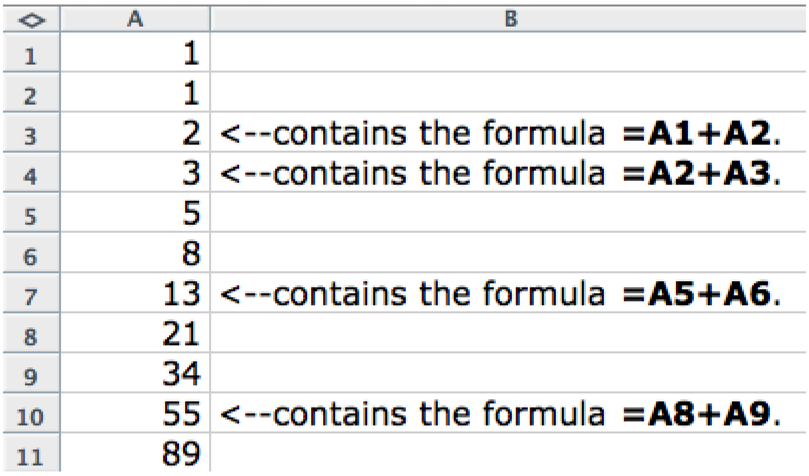
Most of the examples discussed here have been incorporated into discrete mathematics classes and classes for teachers. With the ubiquitous availability of spreadsheets, the author has not encountered a problem with students not having access to the required software. In these classroom applications, there is no sensitivity to operating system or the particular brand of spreadsheet. The rare student who doesn't have a spreadsheet installed on his or her computer can download *Open Office* or *NeoOffice*, or use an online spreadsheet, as in *Google Documents*. The examples given here were analyzed using *Excel* 2008 on a MAC computer (OS X). This paper assumes some familiarity with the spreadsheet and includes only a rudimentary introduction, through example, to the details of entering formulas.

The emphasis here is on using the spreadsheet to analyze recurrence relations. Examples of the spreadsheet's utility in other educational contexts abound. The electronic journal *Spreadsheets in Education (eJSiE)* is a rich source for these applications.

2 A Simple Example

This section uses the Fibonacci sequence, $a_1 = a_2 = 1$, $a_n = a_{n-1} + a_{n-2}$ for $n \geq 3$, as an example to illustrate how a recurrence relation is encoded in a spreadsheet. This section can be skipped if you are familiar with spreadsheet basics.

The contents, data and formulas, of a spreadsheet are contained in cells arranged in a two-dimensional *data array*. The columns are identified with letters, and the rows are identified with numbers. Letter-number pairs are the coordinates of a specific cell, for example **`B3`** identifies the cell in the second column and the third row. (Some spreadsheets would give the coordinates of this cell as **`R3C2`**, row 3, column 2.) The content of a cell is either a constant or a formula based on the contents of other cells. The spreadsheet displays the value stored in each cell, either the constant or the result of applying the formula.



◇	A	B
1	1	
2	1	
3	2	<--contains the formula **=A1+A2**.
4	3	<--contains the formula **=A2+A3**.
5	5	
6	8	
7	13	<--contains the formula **=A5+A6**.
8	21	
9	34	
10	55	<--contains the formula **=A8+A9**.
11	89	

Figure 1: The Fibonacci Sequence

Enter the first two elements of the Fibonacci sequence by typing a 1 in cells **`A1`** and **`A2`**. Now enter the formula **`=A1+A2`** into cell **`A3`**. This is accomplished by selecting cell **`A3`** and entering the formula using one or a combination of two methods: (1) type it in character by character, or (2) type the **`=`**, which indicates the cell contains a formula, select **`A1`** with the mouse, type **`+`**, and then select

A2 with the mouse. Whichever technique is used, the formula builder shows the formula **=A1+A2** as it is constructed. The formula in cell **A3** depends on the values in cells **A1** and **A2**. The data area in the top three cells of column **A** will display the values 1, 1, and 2, the result of the formula **=A1+A2**. The coordinates for a formula seen in the formula builder are the external representation of the cells on which the formula is based. Internally, the formula in **A3** is "The value stored in me (this cell) is the sum of the value stored in the cell two cells above me and the value stored in the cell immediately above me." When the formula is copied to a new cell, it is this internal representation that is copied, and the external representation displayed in the formula builder is adjusted accordingly. For example, selecting cell **A3**, copying its contents, and then pasting into cell **A10** will result in the formula building showing **=A8+A9** as the contents of cell **A10**. Figure 1 illustrates the first few rows of this spreadsheet.

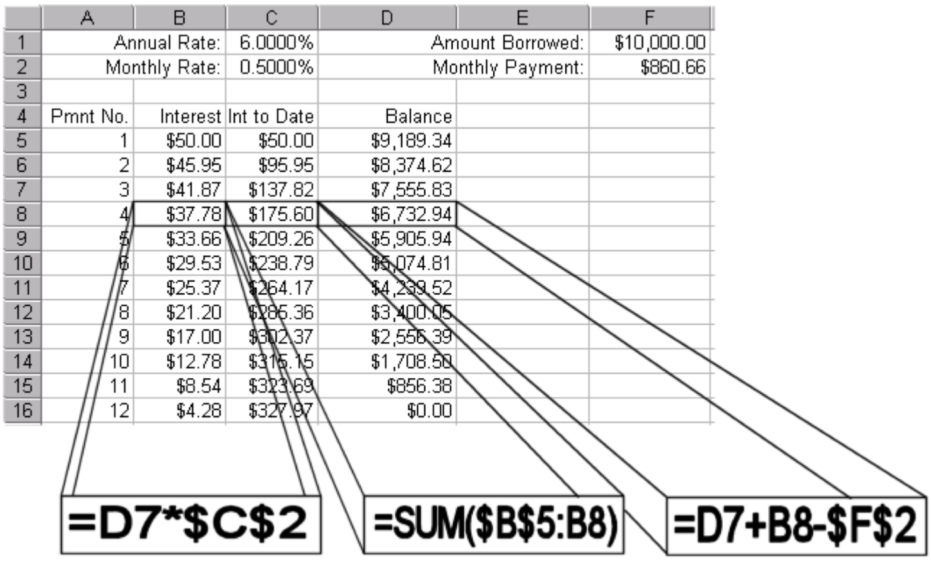
The letter-number coordinates of a cell give its *address*. The user sees in the formula builder the *absolute adresses* of the cells the formula is based on. Internally the spreadsheet stores *relative addresses*, which give the position of the cells in the formula relative to the cell containing the formula. In our example, the relative addresses are "the cell two positions above me" and "the cell immediately above me." It is this relative addressing scheme that simplifies storing recurrence relations in a spreadsheet.

Column **A** contains the first three elements of the Fibonacci sequence. In *Excel*, if you highlight cell **A3**, you will see a small square in the lower right hand corner of the cell. This is a handle that simplifies copying the contents of the cell into other cells. Clicking on the handle and dragging it straight down copies the formula in **A3** into the cells below it in column **A**. In spreadsheets other than *Excel*, you may need to use copy and paste to copy the formula into other cells. After copying the contents of **A3**, the nonempty cells of column **A** contain the elements of the Fibonacci sequence—a numerical solution for the recurrence relation $a_1 = a_2 = 1$, and $a_n = a_{n-1} + a_{n-2}$ for $n \geq 3$, because each cell contains the sum of the two values immediately above it. Notice that it is the relative addressing that encodes the Fibonacci recurrence. The formula in cell **A3** describes its value as the sum of the values in the two cells above. It is this dependence on the two previous cells that is copied into each of the cells below **A3**.

Relative addressing facilitates entering recurrence relations in a spreadsheet, but occasionally it is necessary to specify a fixed cell in a formula. The **$** is used to indicate that the absolute address is stored both internally and externally. For example, if the content of **A3** is the formula **=$A$1+A2**, then copying this formula to **A10** would result in **=$A$1+A9** being copied into that cell. All copies of this formula use the contents of **A1** for the first term. Absolute addressing can be used on each coordinate independently. We will see examples of absolute addressing in the examples that follow. Note that our spreadsheet solution gives us as many terms for the Fibonacci sequence as we want, but it doesn't give us a formula for the sequence. The spreadsheet is a powerful tool for exploring recurrence relations without having to find an explicit solution.

3 A Finance Example

This section contains a problem used in a technology course for licensed teachers. A business pays back a loan of $10,000 over a period of 12 months at an annual interest rate of 6%. What is the monthly payment? Rather than using the spreadsheet's built-in formula for the monthly payment, we



	A	B	C	D	E	F
1		Annual Rate:	6.0000%		Amount Borrowed:	$10,000.00
2		Monthly Rate:	0.5000%		Monthly Payment:	$860.66
3						
4	Pmnt No.	Interest	Int to Date	Balance		
5	1	$50.00	$50.00	$9,189.34		
6	2	$45.95	$95.95	$8,374.62		
7	3	$41.87	$137.82	$7,555.83		
8	4	$37.78	$175.60	$6,732.94		
9	5	$33.66	$209.26	$5,905.94		
10	6	$29.53	$238.79	$5,074.81		
11	7	$25.37	$264.17	$4,239.52		
12	8	$21.20	$285.36	$3,400.05		
13	9	$17.00	$302.37	$2,556.39		
14	10	$12.78	$315.15	$1,708.50		
15	11	$8.54	$323.69	$856.38		
16	12	$4.28	$327.97	$0.00		

=D7*$C$2

=SUM($B$5:B8)

=D7+B8-$F$2

Figure 2: Loan Spreadsheet

develop a recursive model that will lead to the solution and incorporate that model into a spreadsheet that can be used to illustrate the derivation of the formula. Students use a worksheet and, through a series of "leading questions," are guided to a recurrence relation with boundary conditions. Letting $B(n)$ denote the balance owed after the n^{th} payment, r the monthly interest rate, and M the monthly payment, then "the balance after the n^{th} payment equals the balance after the $(n-1)^{\text{st}}$ payment plus the interest earned on the previous balance minus the monthly payment" is encoded as $B(n) = (1+r)B(n-1) - M$. The boundary conditions are $B(0) = \$10,000$ and $B(12) = \$0.00$.

The spreadsheet in Figure 2 encodes the recurrence for B. The boundary condition $B(0) = \$10,000$ is entered as a constant in cell **F1** . The formula in cell **B8** calculates the the interest charged on the previous balance (**D7**). The formula in cell **D8** encodes our recurrence: The current balance is the previous balance (**D7**) plus the interest charged (**B8**) minus the monthly payment (**$F$2**). These formulas are entered with the correct coordinates first in cells **B6** . . . **D6**. These cells are then selected and dragged down to populate the rest of the table. Note the use of absolute addressing for the monthly interest rate (**$C$2**) and the monthly payment (**$F$2**). If absolute addressing had not been used here, then when the formulas are dragged down these would refer to cells other than the ones containing the desired values. The formula in cell **C8** is another illustration of absolute addressing and of the spreadsheet function **SUM**. It could also have been entered as **=C7+B8**.

The value in cell **D16** is the result of calculating the balance after the last monthly payment. When the correct monthly payment is entered into cell **F2**, the value displayed in **D16** is the other boundary condition, $B(12) = \$0.00$. The constant in **F2** is initially a guess. Students can manipulate it until the final balance is $0.00, at which point **F2** contains the correct monthly payment.

The spreadsheet, in addition to containing the solution to the problem posed, also contains a complete example of the type of transaction we are considering. Students can refer to this table and validate the intermediate steps encountered in solving for the monthly payment M when a loan of P is repaid with N monthly payments. From the recurrence

$$B(n) = (1+r)B(n-1) - M,$$

it follows that

$$B(n) - B(n-1) = (1+r)(B(n-1) - B(n-2)).$$

Applying this to successive values of n, using $B(1) - B(0) = Pr - M$, yields

$$B(n) - B(n-1) = (Pr - M)(1+r)^{n-1}.$$

Now,

$$B(N) - B(0) = \sum_{n=0}^{N-1}(B(n) - B(n-1)) = (Pr - M)\sum_{n=0}^{N-1}(1+r)^n = (Pr - M)\frac{(1+r)^N - 1}{r}.$$

Using the boundary conditions $B(0) = P$ and $B(N) = 0$ and solving for M yields

$$M = \frac{Pr}{1 - (1+r)^{-N}}.$$

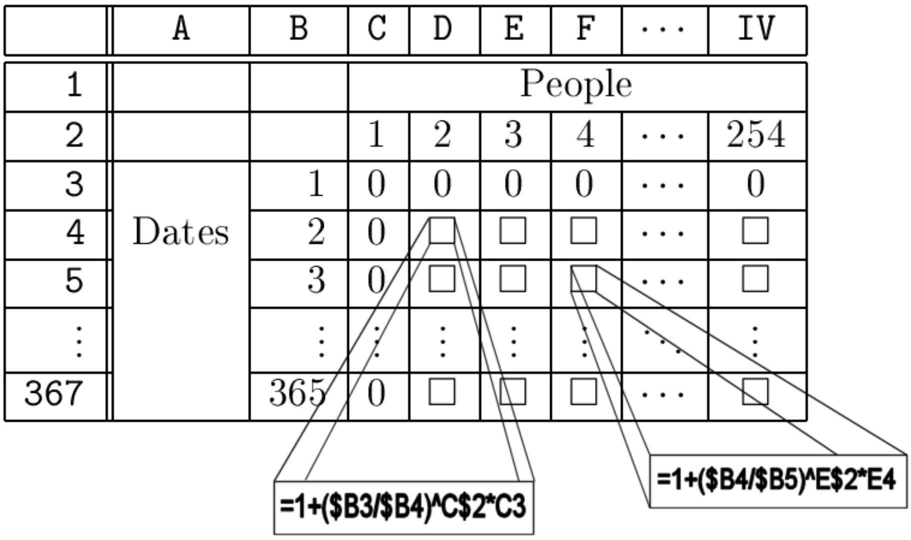
Every stage in the derivation of M can be related to the complete example contained in the spreadsheet. Additional columns, *e.g.* for $B(n) - B(n-1)$, $(1+r)(B(n-1) - B(n-2))$, and $(Pr - M)(1+r)^{n-1}$, can be added to the spreadsheet for the purpose of validating the intermediate steps in the derivation of the formula for M. Thus the spreadsheet, in addition to containing the complete solution for the original problem and illustrating the application of the recurrence relation, supports the student during the analytical solution to the problem.

The main benefit of this approach is that students can develop a recursive model and complete a fairly extensive analysis of it before learning the analytical techniques for solving recurrence relations. This helps (1) to develop the skill of thinking recursively about problems that lend themselves to solution by recurrence relations and (2) to motivate the study of recurrence relations. Although students will learn techniques for solving difference equations analytically, they have seen how the spreadsheet is a powerful tool that can yield solutions without having to derive a formula from the model—a tool that can be used when finding a solution analytically is difficult or impossible. More illustrative examples follow.

The loan payment calculation is an example for which the spreadsheet provides a numerical solution to the problem but also for which an analytical solution is possible. The next section gives an example of a problem that does not have an easy analytical solution.

4 The Birthday Problem

This problem arose serendipitously in one of my Finite Mathematics classes. A standard example in Finite Mathematics is the derivation of the formula for p_n, the probability that, in a group of n people, at least two of them have the same birthday. For groups with 23 or more people, the probability of a birthday match is greater than 50%. My class of 30 students ($p_{30} \approx 0.706$) decided to check out this surprising result. One by one we announced our birthdays and found a match after only seven announcements. My students were surprised that it took so few tries, and we wondered if this is an anomalous result or close to what is expected. This prompted the following question: “In a group of n people, what is the expected number of queries (birthday announcements) required to find a match



	A	B	C	D	E	F	···	IV
1			People					
2			1	2	3	4	···	254
3	Dates	1	0	0	0	0	···	0
4		2	0	□	□	□	···	□
5		3	0	□	□	□	···	□
⋮		⋮	⋮	⋮	⋮	⋮	⋱	⋮
367		365	0	□	□	□	···	□

=1+($B3/$B4)^C$2*C3

=1+($B4/$B5)^E$2*E4

Figure 3: Birthday Problem Spreadsheet

or determine that one does not exist?" This problem lends itself to a spreadsheet analysis. Although my class did not conduct the analysis, they had little trouble understanding the recurrence relation that I used, and those who had used a spreadsheet before could see how it was used to analyze the recurrence.

Our analysis begins with a definition. Let $E(n, m)$ denote the average number of queries required to find a match among n people who are known to have birthdays among a set of m possible dates, or to determine that there is no match. Our goal is to find $E(30, 365)$. The boundary conditions $E(n, 1) = E(1, m) = 0$ for all $n, m \geq 1$ are easily justified. If n people are known to have birthdays in a set containing only one date, then we know without making any queries that a match exists if $n > 1$ or doesn't exist if $n = 1$. If there is only one person in the group, then regardless of the number of possible dates, we know without having to make a query that there is not a match.

Our recurrence is

$$E(n, m) = 1 + \left(\frac{m-1}{m}\right)^{n-1} E(n-1, m-1), \text{ for all } n \geq 2 \text{ and } m \geq 2. \quad (1)$$

The 1 appears in this formula because at least one query will be required if there is more than one person with birthdays in a set with more than one date. The factor $\left(\frac{m-1}{m}\right)^{n-1}$ is the probability that the remaining $n - 1$ people have birthdays different from that of the person making the announcement, and it is with this probability that we need to continue the search with $n - 1$ people who are known to have birthdays in a set with $m - 1$ days. $E(n - 1, m - 1)$ is the average number of queries needed in the event that the current announcement does not find a match.

Figure 3 is a representation of the spreadsheet for solving this recurrence. The zeroes in row **3** and column **C** record the boundary values $E(n, 1) = E(1, m) = 0$. The squares indicate cells in which a formula is stored. Each of these formulas, as for the ones depicted for cells **D4** and **F5**, relies on the number of dates for the previous row, the number of dates for the current row, the number of people for the previous column and the calculated average value for the previous row and column in exactly the manner prescribed by (1). Only the formula in cell **D4** is entered "manually." Selecting this cell and dragging to the right enters all the formulas in row **4**. Selecting the formulas in row **4** and dragging down populates the table, at which point the numeric values for $E(m, n)$ are available in the table. Observe that the entire array of formulas (92,092 cells) is populated with two dragging operations.

The answer to our question is $E(30, 365) \approx 14.289$, found in cell **AF367**. This is the average number of queries needed to find a birthday match, or determine that one does not exist, in a group of 30 people. Since the probability of a match is $p_{30} \approx 0.706$, this calculated average includes the 29 queries that will be needed in the 29.4% of the cases when there is no match. It follows that the average number of queries needed when there is a match is approximately 8.163—so our classroom experience is not out of line with what is expected.

We can also find, using the spreadsheet, the probability mass function (*pmf*) $p(t)$, the probability that t queries are needed to find a match in a group with N people, $N \leq 365$, or to determine that there is not a match. Since, if there is a match, it will be found in at most $N-1$ queries, $p(t)$ is the probability of finding the match in exactly t queries when $1 \leq t \leq N-1$, and $p(N)$ is the probability that there is no match (which will require exactly $N-1$ queries to determine). Letting $p(0) = 0$, the domain of p can be taken to be $0 \leq t \leq N$. Let $P(t) = \sum_{i=0}^{t} p(i)$ be the cumulative distribution function. Then $P(0) = 0$, $P(N) = 1$, and P(t) satisfies the recurrence

$$P(t) = 1 - \left(\frac{365-t}{365-t+1}\right)^{N-t} (1 - P(t-1)) \text{ for } 1 \leq t \leq N-1. \tag{2}$$

As justification, note that $1 - P(t-1)$ is the probability that t or more queries (announcements) are required, and $\left(\frac{365-t}{365-t+1}\right)^{N-t}$ is the probability that the last $N-t$ people have birthdays different from that announced by the t^{th} person. Therefore $\left(\frac{365-t}{365-t+1}\right)^{N-t}(1 - P(t-1))$ is the probability that t or more queries are required and the t^{th} query fails to produce a match. The complementary probability $1 - \left(\frac{365-t}{365-t+1}\right)^{N-t}(1 - P(t-1))$ is thus for the event that fewer than t queries are required or that the t^{th} query succeeds, *i.e.* is equal to $P(t)$.

The recurrence in (2) can be encoded for, say $N = 30$, in a single column of a spreadsheet, giving a tabular representation for the cumulative distribution function P. The pmf p can be encoded in an adjacent column using the formula $p(t) = P(t) - P(t-1)$, for $1 \leq t \leq 30$, and finally, a third column can be used to calculate the expectation $E(30, 365) = \sum_{t=1}^{29} t\, p(t) + 29p(30)$. Consistent with our earlier analysis, the calculated expectation is $E(30, 365) \approx 14.289$.

More details about this problem, together with the results of simulations that validate the results and discussions relating this problem to other problems in computer science and the analysis of algorithms can be found in [2].

5 Tossing Coins

This section describes a problem used in one of our professional development classes for secondary teachers. The students solve the problem by encoding recurrence relations into a spreadsheet. Here is the problem. Two players, Tom and Jim, play a coin tossing game. Tom begins with 13 pennies and Jim begins with 27 pennies. A fair coin is flipped. If it comes up heads, Tom wins one of Jim's

	A	**B**	**C**	**D**	$\cdots$	**AO**	**AP**
1	Number of Coins	0	1	2	$\cdots$	39	40
2	Probability of Winning	0	0.1		$\cdots$		
3					$\cdots$		
4	Expected Number of Coin Flips	0	20		$\cdots$		

Figure 4: Spreadsheet for Coin Tossing Problem

pennies. If it comes up tails, Jim wins one of Tom's pennies. The process is repeated until one player wins all of the pennies. Answer the following questions: (1) What is the probability that Tom wins all the pennies? (2) What is the probability that Jim wins all the pennies? (3) What is the probability that the game will not terminate? (4) What is the expected number of coin flips until the game ends? (*i.e.* If games were played many times from the same starting condition, what, in number of coin flips, would the average length of the games be?)

In class, an analysis is accomplished with a worksheet that asks a sequence of "leading questions" and directs students to recursive models that can answer the questions. We begin by defining some functions. Let $p(n)$, for $0 \leq n \leq 40$ denote the probability that a player with n pennies ends up winning the game, and let $l(n)$ denote the expected number of coin tosses to end the game when a specific player (say Tom) has n coins. Then $p(0) = 0$, $p(40) = 1$, and for $0 < k < 40$, $p(k) = \frac{1}{2}p(k-1) + \frac{1}{2}p(k+1)$, or $p(k+1) = 2p(k) - p(k-1)$. Students develop and justify this recurrence on their worksheets. For l, we reason that $l(0) = l(40) = 0$, and, for $0 < k < 40$, $l(k) = 1 + \frac{1}{2}l(k-1) + \frac{1}{2}l(k+1)$, or $l(k+1) = 2l(k) - l(k-1) - 2$.

To solve the recursive models for p and l, students are asked to make a spreadsheet like the one in Figure 4. The entries in **C2** and **C4** are constants—not formulas. The entries in **B2** and **B4** encode the boundary conditions $p(0) = 0$ and $l(0) = 0$, respectively. Now, encode the recurrence $p(k+1) = 2p(k) - p(k-1)$ by installing the formula **=2*C2-B2** into **D2** and then dragging it into cells **E2**...**AP2** (so that, for example, **M2** now contains the formula **=2*L2-K2**). If the value displayed in **AP2** is not the other boundary condition, $p(40) = 1$, then the constant entered in **C2** is not correct. Students are instructed to manipulate the value in **C2** until the value 1 appears in cell **AP2**. When the correct value (0.025) is found for **C2**$(= p(1))$, the spreadsheet contains a complete tabular representation for $p(n)$. Similarly, the recurrence $l(k+1) = 2l(k) - l(k-1) - 2$ can be installed in **D4**...**AP4** by entering the formula **=2*C4-B4-2** into **D4** and then dragging it into the remaining cells. The value in **C4** is manipulated until the value in **AP4** matches the boundary condition $l(40) = 0$. This occurs when the value in **C4** is $39 = l(1)$. The spreadsheet now contains a tabular representation of $l(n)$.

We can now read from the spreadsheet the answers to our questions. The probability that Tom wins all the pennies is $p(13) = 0.325$, and the probability that Jim wins them all is $p(27) = 0.675$. Since these add up to 1, the probability that the game does not terminate is 0. The expected length of the game is $l(13) = 351$.

When the spreadsheet contains the correct recurrence relations and the boundary conditions are satisfied, the formulas for $p(n)$ and $l(n)$ can be found by inspection. With a little reflection, most students can see that the formulas are $p(n) = n/40$ and $l(n) = n(40 - n)$. That we can discover

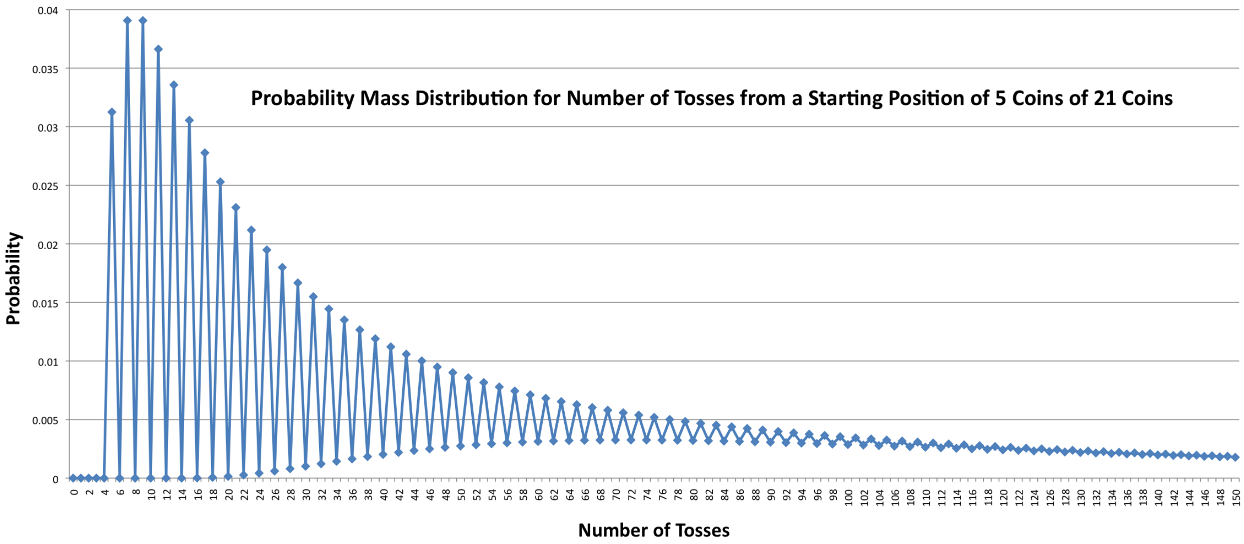


Figure 5: Probability Distribution for Game Length

the formulas is a bonus. Note that the recurrence $p(k+1) = 2p(k) - p(k-1)$ can be rewritten as $p(k+1) - p(k) = p(k) - p(k-1)$, implying a constant difference between consecutive values of p. The formula for $p(n)$ can be found analytically by collapsing the telescoping sum $\sum_1^{40}(p(k) - p(k-1))$ and using the boundary conditions to find the value $(1/40)$ for the common difference $p(k) - p(k-1)$. A similar approach that exploits the symmetry of $l(n)$ can be used to find a formula for $l(n)$. Details can be found in [1] and [3]. The intermediate steps of the analytical solutions can be validated by adding the appropriate formulas into additional lines of the spreadsheet.

Suppose Tom has 5 of 21 coins. As in the previous example, Tom wins or loses a coin based on the toss of a fair coin, and the game is played until Tom loses all of the coins or wins all of the coins. Find the pmf for the number t of tosses required for the game to end. Again, we can set up a recurrence that can be entered into a spreadsheet in order to find a tabular representation of the pmf.

We begin with a definition. Let $q(n, t)$, for $t \geq 0$ and $0 \leq n \leq 21$ denote the probability that the game ends in t moves (tosses of the coin) when Tom has n coins. The boundary conditions $q(0, 0) = q(21, 0) = 1$, $q(n, 0) = 0$ for $1 \leq n \leq 20$, and $q(0, t) = q(21, t) = 0$ for $t \geq 1$ are easily justified. If Tom has n coins, $1 \leq n \leq 20$, and $t \geq 1$, then the probability that the game ends in t moves is the sum of the probabilities for two mutually exclusive events: Either Tom loses a coin and then the game concludes in $t - 1$ more moves, or Tom wins a coin and then the game ends in $t - 1$ more moves. This yields the recurrence

$$q(n, t) = \frac{1}{2}q(n-1, t-1) + \frac{1}{2}q(n+1, t-1). \tag{3}$$

The pmf we seek is $q(5, t)$, $t \geq 0$.

As with the other examples, this recurrence relation with boundary conditions can be encoded in a spreadsheet table. Arrange the table with 22 column headings representing the number of coins held at the beginning of the game $(0, 1, \ldots, 21)$ and with rows labelled with possible values for t $(0, 1, 2, \ldots)$, each row holding the probabilities that the game ends in t moves. Since t can be any nonnegative integer, the desired table has infinitely many rows. Clearly only a finite part can be entered. Enter the boundary conditions in the top row and in the left and right boundary columns, column 0 and column 21—a 1 in each top corner and zeroes elsewhere in the boundary. Now encode the the spreadsheet formula for the recurrence (3) in the cell corresponding to $q(1,1)$ and drag that formula so that the interior 20 columns are populated with the recurrence. (For example, if **D4** is the cell to contain the calculated value of $q(1,1)$, the **D4** contains the formula **=0.5*C3+0.5*E3**.)

The number of rows to populate can be determined with some experimentation. At somewhere around $t = 2000$, all the cumulative sums of the probabilities in each column sum to one, within the precision of my spreadsheet, so the first 2000 or so columns is probably good enough. With the formulas entered into the table, column 5 contains the pmf $q(5,t)$ for the number of moves in a game from a starting position of 5 coins. The spreadsheet's graphics tools can quickly produce a plot of the pmf. Figure 5 shows the spreadsheet graph for $q(5,t)$, $t = 0, 1, 2, \ldots, 150$. The spreadsheet has the capability to remove the lines in the graph, and it would be appropriate to to this, but leaving them in helps to maintain the vertical alignment visually. The graph prompts some discussion questions: Why, for even values of t, do the probabilities that the game will end in t moves tend to be lower than for odd values of t? Why are the probabilities for even values of $t \leq 14$ and for all values ≤ 4 equal to zero? Is there a value for t beyond which the pmf is monotonic?

An analysis similar to the above, with a total of 40 coins in play, indicates that with 21 coins in play, the expected number of moves until the end of the game is $5(21 - 5) = 80$. Using the pmf to calculate the expected value should therefore yield $\sum_{t=0}^{\infty} t\, q(5,t) = 80$. An additional column in the spreadsheet can be added to calculate the partial sum $\sum_{t=0}^{2000} t\, q(5,t) = 79.9999997$, a result consistent with our earlier analysis.

6 Conclusion

The spreadsheet is a powerful computational tool for exploring discrete models. Its computational and graphical capabilities allow the investigator to explore the characteristics of a model that may be difficult or impossible to solve analytically. In all of the examples given here, the emphasis has been on developing a recursive model and exploring and analyzing that model with a spreadsheet. By applying these techniques, students develop a facility for discrete modeling and can include the electronic spreadsheet as one more tool in their problem-solving arsenals.

References

[1] Honsberger, Ross, *Mathematical Gems I*, The Mathematical Association of America (1973), pp 128–130.

[2] McMillan, Thomas C. "The Birthday Problem: Using an Electronic Spreadsheet in a Gentle Introduction to Discrete Mathematics," *Proceedings of the 2^{nd} International Conference on the Teaching of Mathematics*, CD-ROM link, John Wiley & Sons, Inc., July, 2002.

[3] McMillan, Thomas C. "The Gambler's Ruin–How Long Does It Take?", *The AMATYC Review*, Vol. 23, No. 2, (Spring, 2002), pp 33–38.