

# Chapter 6

## Introducing Computational Thinking Through Spreadsheets



John Sanford

### 6.1 Background

To a large extent culture defines who we are. The human species is gregarious. Humans like to interact with other humans. These interactions follow acceptable protocol. Cultural change is generally a slow process barring cataclysmic occurrences. Computational thinking requires a cultural change. It involves a change in the way humans think of using words and numbers and relationships. Advances in computational education require the educators themselves to think that way, and this alone presents a considerable difficulty.

As prehistoric cultures progressed, they developed methods for recording information and keeping a record of significant occurrences and features of their lives. The earliest such records constituted an oral history. Eventually skills of reading, writing, and arithmetic were acquired by a select few in each community. These skills certainly did not include digital computational thinking because the technology did not exist. The digital computer with all its implications as a problem-solving partner is an extremely recent newcomer to the human cultural scene.

Ancient societies produced some astoundingly intricate and accurate mechanisms as evidenced by the Antikythera mechanism discovered in 1901 of the island of Antikythera. This mechanism, thought to have been constructed in 150 BCE, calculated positions of the sun and moon and provided other celestial data. The abacus, the Roman odometer, water clocks, and similar counting devices certainly exhibited a computational nature. Nonetheless, people at that time would never have considered such instruments as presenting a new way of thinking. They were tools intended to assist humans who were employing existing methods of thinking.

---

J. Sanford (✉)  
Thomas Jefferson University, Philadelphia, PA, USA  
e-mail: [SanfordJ@philau.edu](mailto:SanfordJ@philau.edu)

Digital computer developments of the twentieth century CE created profound differences. Some researchers and practitioners began to use digital computers interactively. The computer became a partner in many areas of research and analysis. This partnership really represented a new paradigm. This was a new way of thinking. People used digital machines to solve problems whose solutions were not available by other means then or now. Pursuit of this human-machine partnership added a new feature to the long established skills of reading, writing, and arithmetic. A somewhat popular though not exclusive term for this added skill is “computational thinking.” Participants in the digital computational field are still struggling to clearly define the concept of computational thinking. Whatever it is, it should become as much a natural activity as writing a note or calculating change for transactions in a department store.

How should the pedagogical community approach the new paradigm? Forecasting the future is a popular pastime but one attended by questionable accuracy. Reviewing the milieu of existing science fiction may be as useful as any other forecasting method. Current developments in the field of artificial intelligence suggest that human-computer cooperation may become extensive. Such a conclusion is not really helpful. It rather clouds the issue of appropriate education choices.

The feeling of urgency may exist, but progress is slow. Many states, cities, and even nations are moving ahead with computer science curriculum. For just a few examples:

- In 2017 the state of Mississippi, USA, will roll out a pilot computer science program in 34 school districts (Wright (2016)).
- San Francisco established a pilot CS program for middle grades in the 2015–2016 school year (Twarek (2016)).
- In 2012 the small nation of Estonia introduced programming for its first graders (Olson (2012)).

The Internet reports quite a few pilot programs intended for instruction of young people in computer coding. Scratch is an online programming language developed at Massachusetts Institute of Technology’s (MIT) Media Lab and is designed to be easy if not intuitive for young children. Children who wish to participate can join a Scratch community to design games and create stories. See <https://scratch.mit.edu/>.

Harvard University offers a Scratch Ed community for educators to join an online community and develop projects for children (<http://scratched.gse.harvard.edu/>).

Minecraft is a popular game programming site for children. Like Scratch, it has its own programming techniques and coding language of sorts (<https://minecraft.net/>).

As noted, these programs are pilot programs and participants self-select, which means they are not necessarily representative of the general population. Such programs are ad hoc attempts to add coding as a separate component to the participants’ education. They are not a part of any standard curricular.

## 6.2 What Is Education?

We are entitled to ask, “What is the purpose of education?” Is education intended as an apprenticeship directed toward useful employment of the adult? Is education intended as a means of propagating the advantages of an existing culture? Is education intended to provide the student with knowledge and skills that will allow the individual to lead a happy and fulfilling life?

The last option is the one that classically applied to those attaining an education at the college level. Frequently the historical goal of college education in the western world was preparation of the youth for service in church and state. In 1860 the United States of America created “land-grant” colleges. The focus of these institutions shifted toward an agricultural and mechanical education. Land-grant colleges were intended to provide an education for all social classes. So perhaps the answer to the question is yes to all three options mentioned in the paragraph above.

Should there be a fundamental educational core for everyone with divergence as the child matures? Some countries have national tests to select students who will go on to higher academic levels and hence to more prestigious and higher-paying occupations. This approach tends to be less widespread today but is still practiced in some societies. Often family wealth is an important consideration.

Today we recognize that the education system should offer all students the same educational opportunities starting with the first grade or even preschool. Diversification may be applicable at higher levels. At least one educational goal should be “to provide the student with knowledge and skills that will allow the individual to lead a happy and fulfilling life.”

With these considerations in mind, we must not allow considerations of computational thinking to diminish emphasis on the skills and knowledge that have been demonstrated in the past as valuable for a fulfilling life in an advanced society. Individuals must develop their own mental capacity for reading, writing, arithmetic, logic, and music. In fact there is evidence that logic and music are important progenitors for problem-solving ability. Students need to read and discuss their readings. Students need to have mastered fundamental arithmetic skills required for normal daily life just as they did before the digital age. We must not allow computational learning to become a siren song to lure the child away from mental agility. And yet we need to introduce computational thinking early in the education process so that it becomes a familiar capability on a par with reading, writing, arithmetic, and other skills.

Children should not become addicted to an electronic calculator or computer. It is easy to use Google for solution of simple mathematical problems that should be done on paper or in the student’s head. Today cell phones possess personal assistants, Siri, Alexa, and Google. More personal assistant apps are available at app stores. Soon these artificial intelligence apps will become all too obliging. The digital assistance is not limited to mathematics. Google and other applications will turn text into audio

and spoken word into text. Translation into foreign languages is readily available. In the near future artificial intelligent applications will provide analysis of paragraphs and even entire books. The future is a little frightening. Students who develop a dependency on these applications too early in life will certainly underdevelop their mental capacity.

So there is the problem. This treatise will recommend early introduction of digital solutions, particularly utilizing spreadsheet technology because it is easy to use and readily available. But that introduction should at an appropriate time.

### 6.3 The Approach to a Solution

There are some well-known guidelines for problem-solving though, as already noted, there are situations in which other approaches are necessary. Generally established approaches are facilitated rather than being replaced by computational thinking methods. The following methodologies are valuable:

- Stating the problem and inherent assumptions
- Systems analysis
- Defining constants and variables (state vector)
- Defining a mathematical model
- Selecting a digital computational mechanism where appropriate
- Utilizing graphical presentations where appropriate

Stating assumptions and listing parameters are a good practice to provide clarity of the task at hand. It is particularly essential in providing repeatability. Other researchers cannot follow or replicate your work if you have not provided all the initial conditions.

Systems analysis involves separation of a problem into separate parts that can be studied individually.

Defining a mathematical model may not apply in every case. Perhaps it should be called a computational model. This model is the framework that defines methodology which will be pursued enroot to the eventual solution.

If we are to use digital computational methods, we must select some application software or some computer coding language. There is indeed a wide assortment from which to choose. Software for use in computers includes more application languages than we have space to list, specialized application software, and even specialized digital machines. A very short partial list of specialized application software would include spreadsheets, Mathcad, Maple, project evaluation and control software, linear programming and goal programming, etc. Discussion on the particular value of spreadsheets will be presented shortly.

## 6.4 What Is Computer Science?

Computer scientists have championed the introduction of computer coding for all elementary and intermediate students. As previously noted, some school districts are introducing computer coding in early grades, sometimes very early grades. In fact, computer scientists have been championing this approach for some time although as suggested by Denning, computational thinking did not originate with computer scientists but rather with the various scientific fields, Denning (2017).

Computer coding teaches a valuable approach to logical thinking. Students writing computer code in whatever language must clearly specify variables and constants that will be part of the problem. They learn to deal with sequential logic and predicate logic, (if... then... else). The risk is that the student will spend more emphasis on language structure and less on problem structure. One popular approach for teaching coding to young people involves having them create games. Computer games are popular, so they will capture the child's interest and at the same time will provide valuable coding experience. But computer code does not directly display itself in the algebraic form typically employed in introductory texts on arithmetic or mathematics. Solutions using computer code may appear more as an adjunct than as natural partner to classical educational methods. Some students develop a natural proclivity for coding just as some naturally have exceptional ability with music and musical instruments. But other students find quite the opposite and for them learning computer code proves to be a difficult task.

Harvard University offers a popular course for computer science majors and nonmajors (CS50X (2017)). The website for the course suggests that it provides:

- “A broad and robust understanding of computer science and programming.
- How to think algorithmically and solve programming problems efficiently.
- Concepts like abstraction, algorithms, data structures, encapsulation, resource management, security, software engineering, and web development.
- Familiarity in a number of languages, including C, Python, SQL, and JavaScript plus CSS and HTML.”

This must be good description of computer science or at least an introduction to it. Computer science as described here is very much involved with computer language and structures. But in order to deal with the problem-solving environment of business or profession, one should also have familiarity and experience with systems analysis, stating inherent assumptions, etc., as was discussed previously. The goal is to think of the digital process as part of the solution methodology and not as an add-on feature. Coding and programming are different from the usual methods encountered in precollege education, and they may well appear to be existing in a separate silo.

## 6.5 What Is Computational Thinking?

What is computational thinking? This is a good question to ask. It is not easy to answer. Much has already been written on the subject. One of the early and significant discussions was given in 2006 by [Jeannette M. Wing of Carnegie Mellon School of Computer Science](#) (Wing (2006)). See also Sanford (2013). As is often the case when attempting to delineate first occurrences, even earlier discussions of the topic are to be found, and no attempt to present an accurate history will be made here.

In his recent article on Computational Thinking in Science, Peter Denning commented that, “Scientists who used computers found themselves routinely designing new ways to advance science. They became computational designers as well as experimenters and theoreticians” (Denning (2017)). He describes the process as one of the solutions comprised of computational steps. But as he points out, these steps may or may not be reducible to analytic algorithms.

Sometimes problem-solvers may obtain solutions through digital computation and yet have no idea how the solutions were actually obtained. Such solutions might involve neural networks, genetic algorithms, crowd sourcing, or other methods known or yet to be discovered. If there is a distinguishing factor to computational thinking, it must be that it is a partnership of human mind and digital machinery. One goal for education, even early childhood education, should be to prepare future researchers to use this partnership. Students should think of digital assistance as just part of a natural process as common as using a pencil or a reference book. And this is computational thinking. Too bad that there is no single word, such as informatics, that would present a good description. In fact, informatics already has a well-known connotation that does not meet the requirement.

The student or researcher needs to think of the problem solution as information in the Claude Shannon sense (Shannon (1949)). In that sense one will operate on the known data to develop a solution. In some cases it is just natural that the path to this solution will involve digital computation. We want to be mindful that the path does not destroy information. The paths to problem solutions should, as much as possible, preserve methods, allow for expansion, clearly delineate assumptions, and preserve repeatability. Students should be introduced to this methodology early in their educational career and in such a way that it augments and does not inhibit essential mental development associated with reading, writing, mathematics, music, science, and logic.

## 6.6 What Is Important?

So, the pedagogical concern with computational thinking really refers to electronic digital computational thinking and to a partnership of human mind and digital machinery. This partnership should become second nature. The student and/or the researcher should think of using digital machinery as readily as one might think of

using a pencil and paper. But, most of us think of digital computation as an add-on not as an integral part of problem-solving. Most young people today grow up using cell phones, social media, and even electronic calculators but do not readily turn to digital machinery for assistance in adjusting recipes for an afternoon picnic or budgeting vacation expenses. If computing machinery is used as an assistant in such activities, the solutions become dynamic. Information is readily available concerning the effect of varying different parameters. Most importantly, if digital assistance is used in small things, then it will naturally become an integral part of large things.

The best way for a classroom instructor to present digital computation as a partnership of human mind with digital machines would be to go through the textbook and provide teaching examples to accompany any topic and every topic where such examples could be applicable. Students should work at least one example. This presents digital solutions as strongly associated with the textbook material rather than some add-on feature.

It is not appropriate to present here material from a particular copyrighted textbook in order to develop computer examples that would couple with it. However, verbal descriptions often do not convey a concept as completely as a picture. So beginning with the next section, we present randomly selected examples of computer solutions that might fit with typical educational topics in elementary and intermediate classes. Keep in mind that these examples are only part of the picture. The other part is the textbook or lecture material that is supported by the computer examples. Tied together they constitute instruction in computational thinking.

Many people think that they are using computational thinking if they use an existing program to find answers that apply only to a fixed set of data. The fullest meaning of computational thinking applies to a way of thinking that prepares a mathematical model which is appropriate for digital machinery. The result is dynamic. It may be used to evaluate variations in result that would be expected from variations in input data. The examples presented in this chapter emphasize this.

There is a large collection of software available from which to choose for the examples just discussed. This includes search software, plotting software, and calculating software. Some applications will allow users to write equations “free form” much as they would with pencil and paper. This chapter suggests the use of spreadsheets. It presents support for this choice. However the field of software development is progressing rapidly, and other excellent choices may be available by the time you read these words.

## 6.7 Why Spreadsheets?

It is easy to say that spreadsheet software can be used in ways that resemble a pencil and paper approach with the added advantages that words are typewritten and mathematical calculations are done for you instead of separately on a calculator. It is easy to say that the computer solution on a spreadsheet can be developed and

displayed in a way parallel to the familiar solution printed in a textbook and as such will be easy for students to visualize. But, "Seeing is believing!" The following pages present a succession of examples intended to demonstrate the applicability of this medium throughout the precollege educational experience. Discussion of more advanced applications can be found elsewhere.

These examples are in no way unique. They do not present any particularly original concepts. In fact their simplicity and commonality are the hallmark of the argument for their introduction as a complimentary feature. The examples here are only a few. Spreadsheet software can be used for just about any problem in precollege mathematics, physics, and science. Spreadsheets update all formulas every time a change is made rather than execute a sequence of instruction as indicated in computer programming. This means that the "Do Loop" is not applicable. Difference equations (or time delays) can generally be implemented by assigning time to successive columns or successive rows. All the other basic programming features such as sequential logic, predicate logic, and even table lookup are available in spreadsheet. In addition they contain a wealth of special features such as finding maximums, minimums, sorting, and others. There will be instances where the spreadsheet is not a valid replacement problem-specific software. Such instances might include specialized tutorial software, software to simulate chemical processes, software for advanced mathematical concepts, etc.

Spreadsheets are widely used for business applications, and knowledge of their capabilities is an asset for knowledge workers.

Descriptions presented in the following pages are definitely not intended as a tutorial. People who wish to learn more about development of spreadsheet methodologies can refer to any number of simple introductory books or even to free material on the Internet.

Remember the ultimate goal is for students to become comfortable with organizing problems in a particular way that will facilitate a computer solution. The student should be comfortable using the computer as an integral part of problem solutions.

Spreadsheets are very visual and as such almost self-teaching (well not quite). They enjoy a wide range of applications in business today and will almost certainly continue to do so in the future. Elementary school teachers may already be familiar with this application for recording grades, seating assignments, and countless other uses.

Microsoft is the leading purveyor of spreadsheet software. It is part of their office suite. And, several open-source office suites containing spreadsheet software are available. All offer similar functionality. In addition they have a macro language that may be used by older students to create stand-alone computer programs. Microsoft Excel uses Visual Basic. This is not the currently preferred language for coding instruction, but if a student has finished an introductory class in some other language, such as Python, Visual Basic will not appear difficult.



The following list itemizes some of the arguments for early introduction of spreadsheets:

1. They are almost as visual as using paper and pencil.
2. Students can begin to produce useful material with minimal instruction.
3. They have an extensive library of easily used features and functions.
4. Graphical presentations are easily produced.
5. It is highly unlikely that they will be superseded any time soon.

So when should students first encounter a spreadsheet?

They begin learning addition, multiplication, and division in the early grades. There is no doubt that students should still memorize addition, subtraction, multiplication, and division “tables” so that they are able to perform common numerical computations mentally. Although calculators are all too available, they have no place for simple calculations. It would be a mistake to use digital aids on any occasion where the problem is easily done without them. Like most other features of the human body, the brain will benefit from exercise.

At the proper time students can bring their arithmetic knowledge to a spreadsheet application. A small amount of overhead instruction is necessary but not arduous. How do you initiate a spreadsheet? How do you save it when you are through with it? How do you open a previously saved spreadsheet? Students will probably already be familiar with such processes because they will have been playing computer games. Also, how do you erase? How do you copy?

The next step is the art of creating functions or equations. At the start these will be only simple equations like  $20 = 5 \times 4$ . The computer will use an asterisk instead of the familiar multiply sign and a slash instead of the familiar divide sign. And that is enough for a start. The rest is developed as the need arises.

Figure 6.1 shows a sample. The student writes `=5*4` but leaves the 20 for the computer to insert by itself. If the student leaves a space at the start, the spreadsheet shows the formula. If the student does not leave a space, the computer shows the correct answer. The student can experiment with this sort of calculation in order to become comfortable with it.

Can this be made more interesting? Suppose we develop an arithmetic game. True enough, many computer games are already available to teach arithmetic. But this game the student will create by herself.

**Fig. 6.1** When writing equations, type a space first, and see the formula (equation). Leave no space and see the computer’s answer as shown in column C

	A	B	C
1		My Name	
2		space	no space
3		= 5*4	20
4		= 4 + 4 + 4	12
5		=4/2	2
6		=456/32	14.25
7			
8			

## 6.8 Where Is Computational Thinking?

The above example uses a spreadsheet to accomplish the same thing that could be accomplished with pencil and paper. However it demonstrates the ability of a computer solution to extend the problem as is done where long division is easily added.

In early grades the student would not yet have been introduced to long division. Division of 456 by 32 could be effected with an electronic calculator, and students will probably have such an app on their cell phones. They should look for that on their cell phones. All of this sharing of computer and problem-solving inculcates what we call computational thinking.

Each and every one of the examples presented in this chapter inculcates computational thinking when it is presented along with standard textbook or lecture material. They exemplify the partnership of the human mind and electronic computing to solve problems. Without a computer such problems would be solved with paper and pencil and, in some cases, with the aid of an electronic calculator. The computer approach presents a dynamic solution where the effect of variations in parameters can be easily examined and where results can be easily presented as a chart if applicable.

## 6.9 An Arithmetic Game with a Spreadsheet

Figure 6.2 shows how the “game” will appear when we are done. Construction is quite simple. The student will type in the labels and then type the numbers in column B. After placing the first two numbers, the student can select those same numbers

**Fig. 6.2** The multiplication game  
This figure shows the spreadsheet as it will appear when completed. The student has partially filled in column D

	A	B	D	E
1		My Name		
2		Lets learn multiplication		
3		multiply by		
4		this number	3	
5		0	0.00	TRUE
6		1	3.00	TRUE
7		2	6.00	TRUE
8		3	9.00	TRUE
9		4	12.00	TRUE
10		5	15.00	TRUE
11		6	18.00	TRUE
12		7		FALSE
13		8		FALSE
14		9		FALSE
15		10		FALSE

together and copy them to the remainder of the column. The computer knows to increase each number according to the example of the first two.

Next the student enters the formula  $= B5*\$D\$4$  into cell C5. The student copies this formula to the bottom of the column. The computer will automatically advance the number parameter, B5 to B6, and B7, and so on, progressively as the formula is copied. The dollar signs lock the parameters that they precede so that they will not change.

The student enters  $= C5 = D5$  in cell E5 and copies this formula to the bottom.

Note that Fig. 6.2 shows column C as very narrow. This column will have the correct product value, and we want to hide it. The column width can be adjusted with the mouse so that it is very narrow. To play the game, the student types the product of 3 and 0 in column D, and the computer places TRUE or FALSE in column E.

Features learned in this spreadsheet are:

1. How to copy numbers to create a sequence
2. How to copy formulas
3. How to type formulas like  $= B5*\$D\$4$  and  $= C5 = D5$
4. Use of \$ for absolute reference
5. How to change the width of a column

That is not too much! Student should be invited to create more games for addition and subtraction but not division. Division presents a problem because the computer uses decimals and not fractions. Decimals are probably a later introduction.

Perhaps it would be fun to have the numbers in column B appear in random order instead of sequentially. We would introduce two new features, the RAND() function and a simple sorting routine. First place  $= RAND()$  in cell A5 and copy it to cell A15. Next sort the range A5:E15 using column A. Figure 6.3 shows a typical result after this step. But when one looks at column A in Fig. 6.3, the numbers do not appear to be in ascending order as we expect them to be. At the completion of the

**Fig. 6.3** The randomized multiplication game

	A	B	D	E
1		My Name		
2		Lets learn multiplication		
3		multiply by		
4		this number	3	
5	0.393	5	15.00	TRUE
6	0.4813	3	9.00	TRUE
7	0.8727	7	21.00	TRUE
8	0.9283	8	24.00	TRUE
9	0.1453	0	0.00	TRUE
10	0.546	1	3.00	TRUE
11	0.9339	4	12.00	TRUE
12	0.3148	6		FALSE
13	0.1954	10		FALSE
14	0.0131	9		FALSE
15	0.9829	2		FALSE

sort operation, the spreadsheet recalculates all functions on the sheet. So then the random numbers are all changed. No matter! The intent was to rearrange the numbers in column B, and this has been accomplished. The final step would be to make column A width very narrow so we would not see the random numbers.

Indeed, every time the student types an answer into a cell in column C, all the random numbers will change, but they will not be seen because the column width has been made small. There are “hide” and “reveal” instructions available, but that would add unnecessary complication. The student has played the game and typed in some answers. Every time a new number is entered, all the RAND() values change. But we don’t care because after the range A5:E15 is sorted, it remains the way it is.

The additional features to be learned are:

1. The RAND() function
2. Sorting

It is important that the students be encouraged to follow this pattern and to create new games. Some student may use imagination and ingenuity to do new things. That would show that they are becoming comfortable with computer use.

## 6.10 A Game to Add Columns of Figures

Another game could add different numbers in a column. We start with three numbers in a column. The student picks them at random and types them into columns as seen in Fig. 6.4. At the top of each column, the student enters the formula to sum the numbers in that column. Figure 6.4 shows the result with the formula just as it is being entered into cell D2. When the student presses ENTER to finish the formula in D2, the letters will disappear, and the proper sum value will appear.

The game is to enter the correct sums for the columns into cells B6, C6, and D6. Then the “FALSE” responses in row 7 will change to “TRUE.” Can the student remember what to type in cell B7? It would be = B2 = B6. Of course the game is not much of a game if the correct answer is visible in row 2. To hide these numbers, change row height. Another trick would be to change the text color to white so it will not be visible.

**Fig. 6.4** Adding columns game

	A	B	C	D	E
1					
2		10	22	=SUM(D2:D5)	
3		3	8	10	
4		4	4	5	
5		3	10	2	
6					
7		FALSE	FALSE	FALSE	

To continue the student will have to change the numbers in the columns by hand. If RAND() is used, the numbers will change every time anything is done to the spreadsheet and nothing will work properly.

## 6.11 Word Problems

Computer use is a perfect adjunct when word problems are introduced in math classes. This is true no matter what software is used. Computational modeling of the problem allows reuse of the same template for other similar problems. This is particularly applicable to spreadsheets because when properly constructed, the sheet contains its own labeling and instructions. The spreadsheet provides a self-contained report and solution at the same time. And often a model will bring to mind new imaginative activities beyond the original scope of the initial problem.

Word problems introduced in the lower grades generally involve simple addition, subtraction, multiplication, and division. For example, a problem might say that Oscar has six apples and Juan has four apples. How many do they have together? If Carla joins them and they give her two apples, how many will Oscar and Juan have left? Such problems can be modeled on a spreadsheet which offers the advantage that it is neat and ordered and provides a pleasing presentation. These simple problems can be done in one's head with little or no formal organization. However they provide learning experience on the computer. Good form dictates that a short restatement of the problem be placed at the top of the sheet. All parameters of the problem should be placed on the sheet, usually at the top left. These constitute what are often referred to as the state vector or state variables. Actually some of them may be constants for the sake of the problem. The state variables do not include universal constants such as  $\pi$  that are available as functions on the computer. But they should include other constants such as the acceleration of gravity that will be approximated because it is actually not a constant and is not a simple calculation.

Figure 6.5 shows a simple practice problem. Here Carla has \$7.50 and Oscar has \$9.00. How much do they have together?

	A	B	C	D	E	F	G
1	Part one: Carla and Oscar are friends. Carla has \$7.50 and Oscar has \$9.00						
2	How much money do they both have together						
3	Carla	\$7.50					
4	Oscar	\$9.00		Carla + Oscar=		\$16.50	
5	Juan	\$8.24					
6							
7	Part two: Juan has \$8.24. if he joins them how much will they all have together						
8							
9				All three =		\$24.74	

Fig. 6.5 Presented for spreadsheet practice

When Juan joins them with \$8.24, how much do all three have together? The student would not create a new sheet for the second part but rather add Juan to the state vector and add part two to the existing sheet. The sheet will contain formulas such as  $=\text{SUM}(B3:B4)$  and  $=\text{SUM}(B3:B5)$ . Students should not do the work in their head and just type the answer. And if the problem changes so that Oscar has \$8.00, the new solutions are obtained by changing the value in the state vector from \$9.00 to \$8.00.

As previously mentioned, the spreadsheet serves as a template for similar problems. The student should be guided to use the same sheet for a problem where Carla has only \$3.50 and Oscar has only \$5.00. New results are obtained by simply changing the initial condition values in the state vector.

Problems like these offer an introduction to spreadsheet capabilities and methods, but they do not suggest the significant advantages that are available through the computer. Eventually these advantages should be introduced. The next example demonstrates array capability and the easy display methods that are available in the spreadsheet.

In this problem, Carla and Oscar are friends. Carla has \$7.50 and Oscar has \$9.00. They decide to pool their resources and buy comic books together. They will share the books. Their favorite comics are “Tales for Zombies” which costs \$3.00 per copy and “Squeaky Crypt” which costs \$2.30 per copy. Because they have not bought any of these for some months, their collections are a bit behind. They will buy the current issues and back issues. So they might buy four “Tales of Zombies,” and then they would have enough money for only one issue of “Squeaky Crypt.” What are the various combinations of issues that they could buy with the money that they have?

Figure 6.6 shows the spreadsheet for this problem. The state vector has a value for Carla + Oscar. Together they have \$16.50. Even though this appears in the area for constants, it should be calculated with a formula. It is not a fundamental data item.

The outer borders of the array show the number of each type of comics purchased. Students can type in the numbers 1, 2, 3, and so on. But they should remember how to type 1 and 2 and then select them and copy. The computer will know that it is to increase each number by one.

Formulas will be used for all the other numbers. The formula in cell C14 is  $=C13*\$B\$8$ . The purpose of each dollar sign is to lock the parameter that follows it. When copying this formula cross the row to cell F14, the  $\$B\$8$  will never change and will always point to the cost of “Tales for Zombies.”

The only tricky formulas are the ones in the center that produce the values 5.30, 8.30, and so on. These numbers show the total dollar cost. So if Carla and Oscar buy one of each of the comic books, they will spend \$5.30. The formula is  $=C\$14 + \$B15$ . Here again the dollar sign locks the parameter that follows it. This allows the student to write the one formula and copy it across and down to achieve the total array. The numbers that are possible without exceeding funds available are shown in bold. Conditional formatting is available to allow the computer to set bold format automatically, but this is an advanced feature. Also an IF function can be used

	A	B	C	D	E	F
1	Carla has \$7.50 and Oscar has \$9.00. They are going to pool					
2	their money and buy comic books. "Tales for Zombies" cost \$3.00 per					
3	copy and "Squeaky Crypt" costs \$2.30 per copy. What are their					
4	purchase options?					
5						
6	Carla	\$7.50				
7	Oscar	\$9.00				
8	Tales	\$3.00				
9	Squeaky	\$2.30				
10	Carla+ Oscar	\$16.50				
11			1, 2, 3, etc. are numbers of issues in that			
12			column or that row.			
13		Tales →	1	2	3	4
14	Squeaky ↓	\$0.00	\$3.00	\$6.00	\$9.00	\$12.00
15	1	\$2.30	<b>5.30</b>	<b>8.30</b>	<b>11.30</b>	<b>14.30</b>
16	2	\$4.60	<b>7.60</b>	<b>10.60</b>	<b>13.60</b>	16.60
17	3	\$6.90	<b>9.90</b>	<b>12.90</b>	<b>15.90</b>	18.90
18	4	\$9.20	<b>12.20</b>	<b>15.20</b>	18.20	21.20
19	5	\$11.50	<b>14.50</b>	17.50	20.50	23.50
20	6	\$13.80	16.80	19.80	22.80	25.80

Fig. 6.6 Showing costs for combinations of comic books

to avoid displaying any numbers that exceed the available funds, but this also is an advanced concept. Most likely the student can set the bold format by hand after inspecting the numbers to see which ones are too large. Perhaps the student may be invited to think of how to add the IF feature after the basic solution is created.

Features learned in this problem:

1. The use of absolute reference, i.e., use of dollar signs in a formula
2. How to set text to bold
3. Maybe how to use an IF function

Students should be encouraged to see how easy it is to obtain new results if Oscar or Carla has a different amount of money or if the cost of a comic book changes. One might ask how the problem would be different if Carla and Oscar had a friend who wished to join in the project.

One might ask how to approach the problem if the selection is to be among three different comic books. If there are three comic books, the array must be three-dimensional. It is not likely that a young student will think of that. Such a problem would be beyond the elementary level. The resultant display would involve a set of two-dimensional arrays, one for each different number of issues of the third comic book. Computer programming code for such a problem would involve three loops. That would be simple, but the display would not differ from the spreadsheet display. It is not easy to display three dimensions on a two-dimensional sheet of paper.

Data is often displayed in arrays. For example, the temperature at noon of every day in a month would be an array. Other examples would be inches of rain each day for a month, student grades on a quiz, and names of students in a class. Sometimes people are interested in the numerical average of the data or the total value of all the data. Sometimes we want to sort the data into alphabetical order. There are ample opportunities for examples and problems.

## 6.12 Student Projects

The intention thus far has been to show how spreadsheets and computer technology can fit as part of more usual instruction rather than as separate stand-alone instruction. After students have developed fundamental facility with spreadsheets, they will enjoy projects that offer more self-expression or even group activities. One example of such a group project might be planning a lemonade stand to raise money for some charity. This is a good project, and there are many lemonade recipes on the Internet. There are even Internet sites that will help with planning.

The solution of any problem should start with the accepted scientific method. First identify all the information that might apply to the problem. What are the physical requirements? What locations are possible? What are advantages of each location? What criteria will be used to select the best location? What hardware such as table, chairs, pitchers, and the like may be required? Does required hardware impact site selection criteria?

People often neglect specification of selection criteria. This oversight may result in selections based more on whim rather than on attributes.

Of course much of the hardware can be borrowed from friends, relatives, or other sources, and these factors are listed as problem assumptions. But students must plan for the expendable items and must establish a price for a cup of lemonade. Modeling the material costs on a spreadsheet will be helpful. It will provide a clear display of all the data and calculations. Further it will allow examining various scenarios that will lead to setting a price and estimating the probable profit of the enterprise.

The first step in modeling the expendables is to obtain a good recipe, preferable from the Internet. Using the Internet rather than human help will encourage resourcefulness and independence. The next step is planning and designing the spreadsheet itself. The various spreadsheets developed by different groups may be similar but will probably differ in appearance. The first part of the sheet should contain a brief description, names, and a date. Then there needs to be places for the recipe, for all the raw materials, and for any constants such as how many cups will be available from a gallon, and so on. When purchases are made, how many units are contained in a package? For example, how many pounds of sugar are in one sack? All numbers need to be entered in separate cells all by themselves with no labels. However, dollar signs are actually part of a number.



	A	B	C	D	E	F	G	H
1	PRELIMINARY PLAN for lemonade stand, Carls & Oscar 4/5/2017							
2	Open Saturday and Sunday afternoons on two weekends.							
3	They will not consider any sales taxes because the money is for a charity.							
4	Borrow table and containers.							
5	One time purchase of cups because they are 1,000 per package.							
6	Refrigerate lemonade unsold on Saturday.							
7	Discard lemonade unsold on Sunday.							
8	One package of ice per day for customers on demand.							
9	Aater chilled overnight so minimal ice will be required.							
10	Estimate 4 gallons per day							
11	After the model is complete, select a sales price to achieve desired profit							
12	after considering what customers might be willing to pay.							
13								
14	RECIPE for ONE GALLON				CONSTANTS			
15	Lemons	(2 cups)	8		Lemons per pound		4	
16	Cups sugar		2		Cups of sugar per pound		2	
17	Cup hot water		1		Ounces per gallon		128	
18	Pints cold water		6.5		ounces per cup (8 oz.)		7	
19					Cups per gallon		18.2857	
20								
21	FIXED COSTS		Units	Cost/Unit			# of Days	TOTALS
22	Cups	pkg.	1000	\$24.49			1	\$24.49
23	Ice	pound	15	\$2.00			4	\$8.00
24								\$32.49
25								
26	VARIABLE COSTS							
27	Item	Sold by	Units	Cost/Unit	Cost/Gallon	Gallons/Day	# of Days	TOTALS
28	sugar	pound	2	\$1.80	\$1.80	4	4	\$28.80
29	lemons	pound	1	\$1.70	\$3.40	4	4	\$54.40
30	water	free			0	4	4	\$0.00
31					\$5.20			\$83.20
32	Total Expense							\$115.69
33								
34	SELLING PRICE CALCULATION							
35	Estimated Sales		units	Price	\$/gallon	gallons/day	# of days	Total
36	Cups		1	\$0.75	\$13.71	4	4	\$219.43
37	Profit							\$103.74

Fig. 6.7 Showing possible lemonade sale configuration

Figure 6.7 shows one possible configuration. Teachers should give some guidelines as a start for the spreadsheet:

1. Begin with a very brief description and include names and the date.
2. Organize in columns with column headings for units, cost per unit, number of days, etc.
3. Think of fixed costs and variable costs separately.

4. Variable costs are costs that depend on how many cups you sell, like lemons and sugar.
5. For raw materials, think of how you buy them, per pound, per unit, etc. and their cost.
6. Set up the spreadsheet so that there are certain numbers entered as numbers such as cost of sugar, cost of ice, number of days, etc.
7. Other numbers should be calculated using a formula that refers to the fixed numbers. Then if you change the cost of lemons, all the other numbers will change (recalculate).

### 6.13 Equations

Consider further use of spreadsheets and computer technology as part of normal in-class instruction in mathematics. Equation (6.1) is a simple example that would have a simple solution:

$$12 = 3X + 3 \tag{6.1}$$

Yes, the solution is  $X = 3$ , but it might be instructive to see the locus of all points defined by  $3X + 3 = Y$ . Figure 6.8 shows this locus. It is easy to produce, and the method is well worth the time it would take a student to learn it. Two columns define

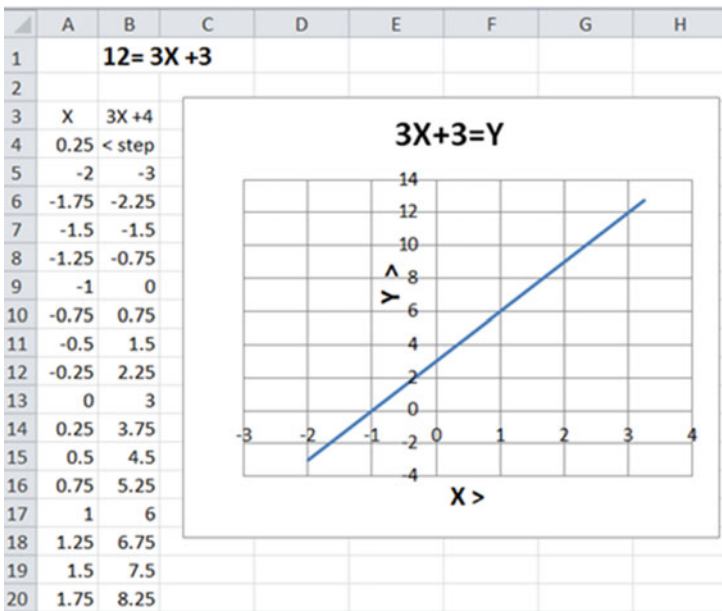


Fig. 6.8 Showing locus of all Y values for  $Y = 3X + 3$

the chart shown in Fig. 6.8. One is a column of X values and the other a column of Y values. Create the columns by selecting a starting value for X and placing it in a cell, in this case cell A5. The value of X should increase by some fixed amount for each cell. This amount of increase can be called the step. Place the step value in cell A4. Now the formula in cell A6 is = A5 + \$A\$4. Copy this down the column for a reasonable way.

The next column has the computation of  $=3X + 3$  obtained by referring to the X value from the corresponding row in the X column. In other words, cell B5 contains  $=3*A5 + 3$ . Copy this formula down the column, and create the chart (graph). All the spreadsheet software contains graphing capabilities although they differ slightly. It is important to use the X-Y plot feature.

Now the student can see that  $Y = 12$  when  $X = 3$ . This is so simple that it might appear trivial, but the technique has broad application. For example, consider Eq. (6.2):

$$3X^2 + 2X - 3 = 0 \quad (6.2)$$

There is, of course, a quadratic formula that may be used to solve this equation for the value of X.

Examples are offered on the Internet. Some schools may have advanced math application software such as Mathcad, Maple, or similar applications that will provide easy methods for finding the solution and even for graphing the function. However, the spreadsheet is included with office suites, and it is essentially free if you own the suite. So we can use it. Figure 6.9 shows the result.

If the student is advanced sufficiently in the use of spreadsheets, values for the equation coefficients, a, b, and c, may be placed at the head of the sheet and referenced in the various formulas where they are required. For a typical example, see Eq. (6.3):

$$= \$B\$1 * A6^2 + \$B\$2 * A6 + \$B\$3 \quad \text{formula in cell B6 of Fig.6.9} \quad (6.3)$$

The solutions are read where the graph intersects the x-axis. Using coefficients in cells B1, B2, and B3 provides the student with the capability of changing the C value and watching the entire curve move up or down. The student can see when the curve has two solutions, only one solution, and when it has no solution (not valid).

Figure 6.9 is a template. It stands ready to deal with any quadratic problem. The only modifications required for a new problem are changing the coefficients and changing the step and start values.

Graphical solutions are also applicable to third-, fourth-, and higher-order equations. As previously mentioned software that will provide such solution is readily available. In fact at least one Internet site is available that will provide roots of a third-order equation. There is still some advantage to using a graphical solution that one creates for oneself. For one thing behavior of the function away from the zero crossing is available. The student can examine sensitivity of the roots to coefficient changes. At what value of the constant term will there be two solutions and only one solution?

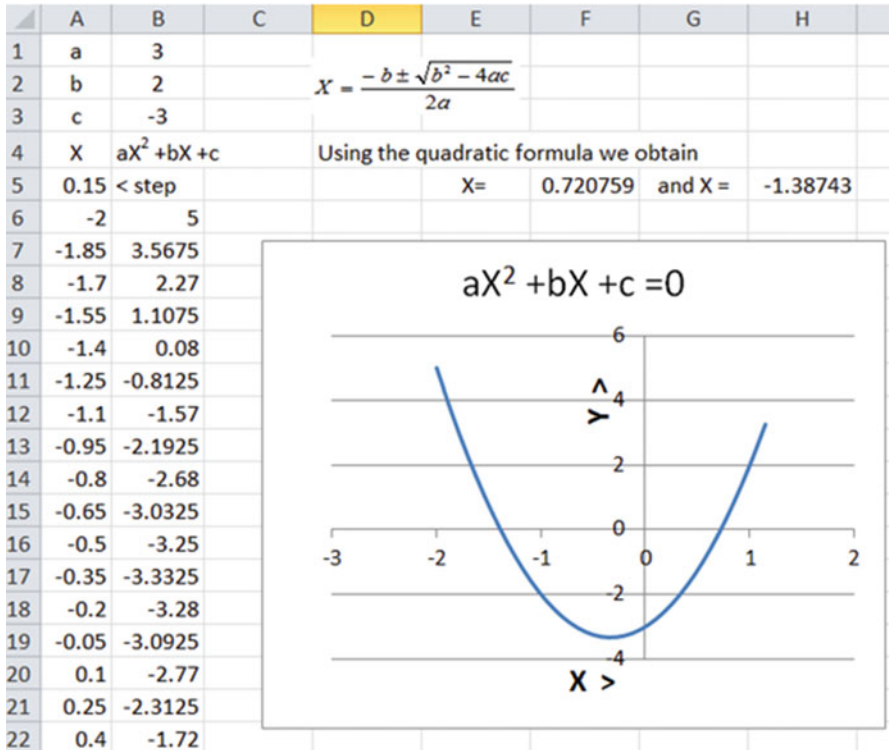


Fig. 6.9 Showing solution of a quadratic equation

As can be seen in Fig. 6.10, the coefficients are listed at the top of the spreadsheet so that they can be readily changed. One can obtain a more accurate determination of the roots by expanding the graph in the area of zero crossings. This requires only that the starting point and step size be changed. A very small step size will expand the area of interest. Figure 6.11 shows this. And, the spreadsheet of Fig. 6.10 is a template for use with any third-order polynomial.

### 6.14 Simulating a Problem in High School General Science

Describing the trajectory of an object thrown up into the air is a common general science problem. It is so common that multiple solutions are available on the Internet. A slight variation would be a rifle bullet trajectory. If the rifle is held at some angle, there will be horizontal and vertical velocities. The vertical component is given by multiplying muzzle velocity by the sin of the angle. For the horizontal component, one uses the cos instead of the sin.

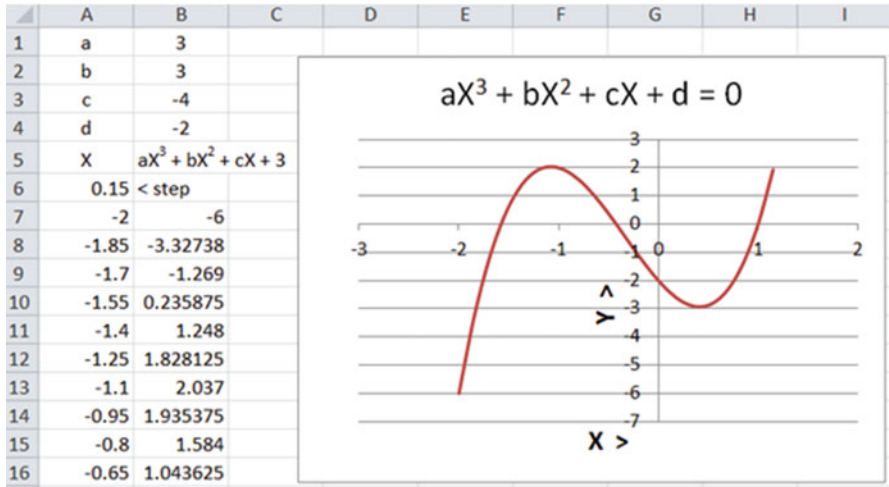


Fig. 6.10 Showing graphical solution of third-order equation

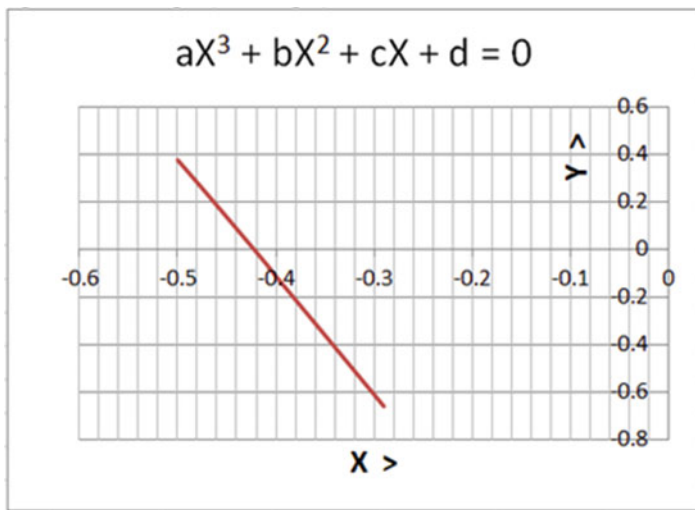


Fig. 6.11 Showing expanded graph in area of the root  $X = -0.423$

Problems like this provide good learning experiences because they have many parts that must be investigated in succession. A spreadsheet allows these components to be displayed in an orderly fashion.

The maximum height of the trajectory occurs when the velocity resulting from gravity equals the upward component of muzzle velocity. The two velocity vectors act in opposite directions. So the time spent traveling up,  $T_U$ , is obtained as the upward component of muzzle velocity divided by the acceleration of gravity. This is

shown in Eq. (6.4). It is the same time that the bullet will require to fall back down to the height of the rifle. The height above the rifle that the bullet will reach in  $T_U$  seconds is given by Eq. (6.5):

$$T_U = V_U/g \quad (6.4)$$

$$D_U = \frac{1}{2}gT_U^2 = V_U T_U \quad (6.5)$$

Total distance from apex to the ground,  $D_T$ , is equal to  $D_U + D_R$ , where  $D_R$  is the distance from the rifle to the ground. The time for the bullet to travel from the apex to the ground is given in Eq. (6.6). So the total time of flight, up and down, is  $T_D + T_U$ :

$$T_D = \sqrt{\frac{2D_T}{g}} \quad (6.6)$$

These equations are derived through use of the calculus, but they are typically presented without derivation in elementary general science textbooks. Some pedagogical assistance is required because the student did not use the calculus to develop the equations and so will not necessarily recognize that the sequence of calculations is important. One must first find the total distance from ground to the apex and then employ Eq. (6.6). Then the time in flight,  $T$ , is the sum of  $T_U$  and  $T_D$  as shown in Eq. (6.7):

$$T = T_U + T_D \quad (6.7)$$

Once the total time of flight is available, the horizontal range can be found as the product of flight time and horizontal component of muzzle velocity, assuming zero air drag.

Figure 6.12 shows atypical spreadsheet solution. This solution utilizes an Excel add-in called “Solver” to select an angle that maximizes the horizontal range.

The Solver add-in is not really necessary. The student can use different values for the angle and optimize heuristically. Students should be encouraged to change initial conditions such as such muzzle velocity and angle to see what effect these have on trajectory.

Plotting the trajectory would be fun if the students are sufficiently familiar with spreadsheets. By the time they reach general science they should be.

As always, the first steps in solving a problem are a clear statement of the problem and a clear statement of parameters (state vector) and influencing factors.

The rifle bullet trajectory problem demonstrates this orderly development. Each of the calculated values is shown in the column for calculated values. Each is calculated using a function that relates to problem constants or other calculated values. The spreadsheet provides an easy method of describing the succession of steps and allows them to be labeled and visually displayed. The spreadsheet provides not only a solution but essentially a report of the solution all packaged together.

All of the values must be related. All formulas must reference the constants in the constants column. Most constants are not laws of nature but are initial conditions that can be changed. Changing any one of them must result in a new solution. Perhaps the

**Fig. 6.12** Showing the bullet trajectory problem

	A	B	C	D
1	NAME Fred	DATE 3/1/2017		
2	Bullet trajectory using gravity at sea level			
3	air drag neglected			
4		Symbol	constants	calculated
5	angle to the horizontal	A	44.976	
6	acceleration of gravity ft/s <sup>2</sup>	g	32.174	
7	muzzle velocity ft/s	V	390	
8	Height of rifle ft.	D <sub>R</sub>	4	
9	feet in a mile	M	5280	
10	upward muzzle velocity	V <sub>U</sub>		275.66
11	Horizontal muzzle velocity	V <sub>H</sub>		275.89
12	Time traveling up	T <sub>U</sub>		8.57
13	Time traveling down	T <sub>D</sub>		8.58
14	Time to hit the ground	T		17.15
15	Max height above rifle ft.	D <sub>U</sub>		1180.86
16	Max total height ft.	D <sub>T</sub>		1184.86
17	Horizontal distance ft.	D <sub>H</sub>		4731.42
18	Horizontal distance miles			0.90
19	Formulas used in the model are shown below.			
20	$V_H = \cos(A) \times V$			
21	$V_U = \sin(A) \times V$			
22	$D_U = \frac{1}{2} g T_U^2 \quad T_D = \sqrt{\frac{2D_T}{g}}$			

most significant risk inherent in spreadsheet solutions is that the problem-solver may perform mental calculations and insert them into the solution process. In that case changing the initial conditions or constants would not produce a correct solution for the new constants.

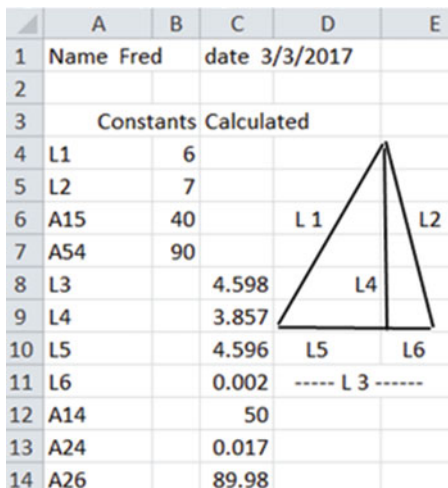
Students will and should display originality in design so all spreadsheets will not all look like Fig. 6.12. Presenting students with a template would defeat the fundamental intention of creating a comfortable relationship with computational thinking. There is little doubt that solutions for this exact same problem will be found on the Internet. Students should resist looking at them before obtaining their own solution unless they are stumped.

### 6.15 Trigonometry

Trigonometry is generally considered to be a later high school subject. Students would typically use calculators in order to obtain required level of accuracy for even simple problems. Built-in functions are available in spreadsheets for both plane and spherical trigonometry and do not require loading special libraries. Simple diagrams can be created right on the spreadsheet using a feature that allows insertion of shapes.

Students still need to know the fundamental rules that are applicable to trigonometry, such as that the sum of all interior angles of a triangle will equal 180° in Euclidian trigonometry. Students will need to know what functions to employ for the

**Fig. 6.13** A typical trigonometric problem



solution to a given problem. The advantage of the spreadsheet over the calculator is that problem visibility is as always present. Changes may easily be made to accommodate new problems that are similar to the given one. Figure 6.13 presents a typical problem.

### 6.16 Simultaneous Algebraic Equations

Introduction of simultaneous equations is a late high school topic. If software similar to Mathcad, Maple, etc. is available, it will present a medium that is easier to use than the spreadsheet. But convenience and expense incline toward use of a spreadsheet. It is there and it is free with any office suite. The spreadsheet will perform matrix operations, addition, subtraction, multiplication, and inversion. Knowledge of matrix operations is valuable for mathematics and for computer science.

Spreadsheets use functions such as MMULT and MINVERSE to accomplish multiplication and inversion for division. These are array operations, and so the input parameters for them are arrays. The easiest way to specify an input array is to select it with the mouse, but one can specify it by identifying the two diagonal cells. Since the response will be an array, one must select the area that the response array will occupy and then type a function such as = MMULT(...

The method for solving simultaneous equations involves the matrix equations shown in (6.8) and (6.9)

where Y, A, and X are matrices. A<sup>-1</sup> is the inverse of A.

$$\{Y\} = [A] \cdot \{X\} \tag{6.8}$$

$$[A]^{-1} \cdot \{Y\} = \{X\} \tag{6.9}$$



**Fig. 6.14** Showing matrix operations

$$\begin{array}{l}
 3 = 3X_1 + 6X_2 \\
 5 = 3X_1 + 8X_2
 \end{array}$$
  

$$\begin{array}{c} \mathbf{Y} \\ \boxed{\begin{array}{c} 3 \\ 5 \end{array}} \end{array} = \begin{array}{c} \mathbf{A} \\ \boxed{\begin{array}{cc} 3 & 6 \\ 2 & 8 \end{array}} \end{array} * \begin{array}{c} \mathbf{X} \\ \boxed{\begin{array}{c} X_1 \\ X_2 \end{array}} \end{array}$$
  

$$\begin{array}{c} \mathbf{A}^{-1} \\ \boxed{\begin{array}{cc} 0.667 & -0.5 \\ -0.17 & 0.3 \end{array}} \end{array} * \begin{array}{c} \mathbf{Y} \\ \boxed{\begin{array}{c} 3 \\ 5 \end{array}} \end{array} = \begin{array}{c} \mathbf{X} \\ \boxed{\begin{array}{c} -0.5 \\ 0.8 \end{array}} \end{array}$$

The X matrix is a column matrix containing  $X_1, X_2, \dots, X_n$ . Actually it is blank in the spreadsheet the labels,  $X_1, X_2$ , etc., are understood. After performing the operation shown in (6.9), X will contain the solution for all values of  $X_1, X_2, \dots, X_n$  dictated by the simultaneous equations. The Y matrix is also a column matrix containing all the Y values that correspond to the rows of the simultaneous equations. The matrix, A, contains the coefficients of all the X values in the simultaneous equations. Figure 6.14 shows this as it might appear in a spreadsheet. The letters Y, A, etc., above the matrices are not necessary. They are added for clarity as are  $X_1$  and  $X_2$  in the initial X matrix. In fact, the entire Y matrix in the lower equation is added for clarity. The original Y matrix can be pre-multiplied by  $A^{-1}$  to yield the answers for X values.

One interesting feature is that the results shown on the spreadsheet are dynamic. If one changes a value in the A matrix or the initial Y matrix, the solution changes to the correct values for the new input. If a changed parameter produces a set of equations for which there is no solution, Excel puts “#NUM!” in the cells. Other spreadsheets may use a different notification.

The simultaneous equations may be many in number, limited only by practical computational considerations. It is valuable for students to know how to solve two simultaneous equations by hand. Certainly more than two simultaneous equations can be solved by hand, but the process does become tedious for large numbers of equations.

## 6.17 Conclusion

This chapter has presented some concepts related to computational thinking.

1. Computational thinking is nothing more than developing a normal tendency for individuals to organize and present problems for solution with the aid of digital computing machinery.
2. It is a partnership of human mind and digital machinery.
3. Digital machinery should be as readily used as paper and pencil.
4. Computational thinking should be incorporated in early childhood education.

5. Computational thinking should be included throughout the education years.
6. It involves a methodology for problem-solving and not just computer programming.
7. Spreadsheet software is suggested as a medium, but there are other options.

Each of the examples shown in the chapter presents a computerized method of problem-solving. When instructors develop similar problems to augment textbook material, they are inculcating computational thinking.

Problem-solving methods such as systems analysis and use of mathematical models are touched on in the chapter. These are not really specific to computational thinking any more than they are components of problem-solving, or computer science, or other disciplines. The chapter has not stressed their existence.

In this world “we must run as fast as we can, just to stay in place” (Carroll). By the time the grade school child of today reaches college age, the human interface with computers will almost certainly be through artificial intelligence and robotics. Does that mean that this concept of computational thinking will fade in significance? No! If humans have a problem, then they must describe the problem even for artificial intelligence. A significant feature present in the examples contained in this chapter is that the first step is describing the problem.

## References

- CS50X. (2017). Harvard University <https://www.edx.org/course/introduction-computer-science-harvardx-cs50x> January 13, 2017.
- Denning, P. J. (2017). Computational thinking in science, American Scientist January 2017, publication of Sigma Xi, January–February 2017 edition pages 13–17.
- Olson, P. (2012). *Why Estonia has started teaching its first-graders to code*. *Forbes.com* <http://www.forbes.com/sites/parmyolson/2012/09/06/why-estonia-has-started-teaching-its-first-graders-to-code/#7de3a61b5790>. Sighted 1/13/2017.
- Sanford, J. F. (2013). Core concepts of computational thinking. *International Journal of Teaching and Case Studies*, 4(1), 1–12.
- Shannon, C. E. (1949). The mathematical theory of communication. ISBN 0-252-72546-8.
- Twarek, B. (2016). *San Francisco Unified School District*, STEM Dept. <http://www.csinsf.org/pilot.html>. Sighted 1/31/2017.
- Wing, M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- Wright, C. M. (2016) (CS4MS) pilot program in 34 Mississippi school districts. <http://www.mde.k12.ms.us/TD/news/2016/04/28/mde-targets-expansion-of-computer-science-education-in-pilot-programs-teacher-training>. Sighted 1/31/2017.

## Other Related Papers by This Author

- Sanford, J. F., & Naidu, J. T. (2016). Computational thinking concepts for grade school. *Contemporary Issues in Education Research*, 9(1), 23.
- Sanford, J. F., & Naidu, J. T. (2017). Mathematical modeling and computational thinking. *Contemporary Issues in Education Research*, 10(2), 159–168.