Math intro program

August 15, 2002

Modeling

Read chapter 3 *Introduction to modeling* in the book *Applied Mathematics: Body & Soul* (AMB&S). If there is something you don't understand, put a question mark in the margin and ask for help to straighten this out in the exercise group or with your teacher. Discuss the *content* of the chapter within the group. What does it say? What does it mean that mathematics is a modelling tool?

Natural numbers

The simplest possible example of a *mathematical model*, in fact, is a (single) *natural number*, like 3 or 12. For example, the number 3 may *model*, or give a "picture of", three bananas, or three elephants, or three attempts to log into a computer system. Read sections 1-3 of chapter 5 in *AMB&S*, about Natural numbers. Again, put a question mark in the margin whenever there is something you don't understand, and be sure to straighten these out as soon as possible together with the rest of the group.

Variables

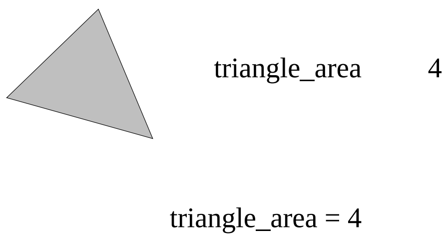
A natural number, like 3, is always "the same", that is, has the same *value*. It is a *constant*. The *interpretation* of 3 (as bananas, elephants, attempts or something else), of course, can vary, but the *value* of 3 is always the same. A *variable*, on the other hand, is something that can take on different values from time to time, like the number of attending students in your exercise group can be 7 one day and 9 another day. The (present) number of attending persons in your group, thus, is a *variable* (hopefully not decreasing with time). Another example is the number of days left to the next exam.

This variable seems to first decrease rather slowly but then faster and faster. Be ready! Give your own examples of variables.

Variable names

In mathematics it is common to give short symbolic one-letter names to variables, such as a, b, f, t and x. There is a lot of tradition in the choice of variable names. For example, you must have noticed that x is a very popular name for a variable whos value is unknown and which you would like to find. When the swedish alphabeth gets exhausted, it is common to continue with the greek letters α, β, γ, δ,.. etc. Here too, tradition is very strong. For example, ϵ almost always denotes a variable with small positive values, like 0.01 or 0.0001.

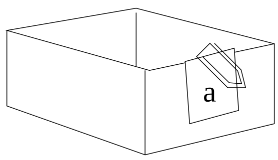
One may also use *words* as names on variables, like *salary*, *shoesize* or *weight*, (variables typically increasing with age). In computer codes it is often useful to give such "suggestive" names to variables, to make the code more readable. Note that variable names may also consist of several words like lastName, triangle_area or number_of_days_left_to_Christmas, for example. Thus, a variable, like the area of a triangle, can be given a *name*, like triangle_area, and carry a *value*, like 4, which we may *model* by writing triangle_area=4.



Variables in a computer

To help us understand how a computer works to keep track of and manipulate variables it is useful to think of the variable as a "box", the *name* of the variable as a "label" on the box, and the "content" of the box as its *value*.

Exercise Make a paper box marked with the letter a (as in the figure below) and put into it 3 matches. We may model, "picture", or denote this as $a = 3$, meaning that the current value of a is 3, or is set to, or defined to be 3. Empty the box, and model this writing $a = 0$. Explain to some of your friends exactly what is going on here.



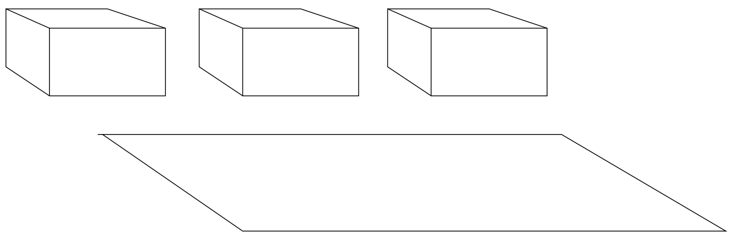
Assigning

The process of putting the three matches into the box a, that is, giving a the value 3, is also refered to as *assigning* a the value 3. To model this one may write $a \leftarrow 3$ or $a := 3$. In this note we adopt the syntax used by Matlab, where this notation is simplified to just $a = 3$. Thus, $a = 3$ does not mean that a "for ever" will carry the value 3, but rather that "at thus very moment" a is given the value 3.

Exercise Discuss in the group the interrelation of the concepts variable, (variable) name and (variable) value. What is meant by assigning a value to a variable?

Building a computer

Make three new paper boxes, corresponding to memory cells of the computer, and put a blank paper on the table corresponding to the computer *screen* for computer output.



Imagine now that on the computer keyboard you type $a = 3$. What happens in the computer is that one of the boxes (memory cells) gets the name a (attach a piece of paper with the text a on the box with a paper-clip), and then the value 3 is *assigned* to a, meaning that a gets the value 3 (put 3 matches in the box so that (the box) a holds the value (number of matches) 3).

Exercise Imagine next that you type $b = 5$ on the keyboard. Perform the corresponding computer action!

If you just type the name of a variable, the computer should respond by displaying the current value of that variable on the screen. For example, if you type b then $b = 5$ is displayed on the screen. Likewise, as you gave a the value 3 before, the computer probably responded by displayed $a = 3$ on the screen, just as an acknowledgement or receipt. Again, what happens if at this point you type (ask for the value of) b? Look into the b-box and write the computers respons on the (paper) screen. It responded $b = 5$, right?

Exercise Discuss in the group what would happen if at this point you type (ask for the value of) c. Hint: Look in the computers memory for a variable labeled c. Do you find one?

Exercise Image next that you now type $c = a$. Show what happens in the computer? Now what would the computer respond if you type (ask for the value of) c.

To check that things really got quite right here, consider what happens if you next type a. Look into the a-box and write the computers respons on the screen. Did it respond with $a = 0$ or $a = 3$? If you think it responded $a = 0$ you should ask the question: who changed the value of a, and then go back and reconsider the action of the computer after you typed $c = a$ (this should not empty a but rather fill c with a *copy* of the content of a).

Assume next that we tell the computer $a = 2$. Explain carefully what happens in the computer. You may come up with two reasonable suggestions; one is that the box a is first emptied and then filled with 2 matches, and the other is that one match is removed from a (recall that the value of a before was 3. The net result, of course, is the same, but the first alternative may be the best exponation because it is independent of the previous state (value) of a.

Assume next that we now type (ask for the value of) c. What does the computer respond? Here you may run into a conflict because some minutes ago we told the computer $c = a$, and a is now 2, but if you look in the box c you find 3 matches, so what is the answer? Discuss in the group this appearent contradiction and what the answer could be. As indicated above, some computer languages uses syntax such as $c := a$ or $c \leftarrow a$ instead of $c = a$ to make it explicit that the (current) value of a is assigned to c, to avoid this conflict. Thus, telling the computer $a = c$ (here meaning assigning the current value of c to a is *not* the same as $c = a$ (which means assigning the value of a to c).

Exercise What would happen if you tell the computer $2 = a$ do you think? Discuss in the group.

Next type $a = a + 3$ and show what happens. Again the computer gets a copy of the content of a (2 (matches)), adds 3 to it, and then *replaces* the content of a by the just computed sum (of 5 matches).

Exercise Explain what happens in the computer as you type $b = 2$, $c = 3$, $b = c + b$.

Matlabs *ans*

Also discuss the computer respons corresponding to typing just $a+2$, without any instruction where to put the result. Matlab has a particular box or variable to put such things called *ans* (for answer). Produce such a box with the name *ans* and fill it with (a copy of) the content of a added with 2.

Exercise What happens in the computer as you type $a = ans + b$, $ans + a$. Show with matches!

Computer "infinities"

Next imagine that you type $d = 2$ on the keyboard. What will the computer do? Probably display something like "out of memory" on the screen. Why?

What would happen if you type $a = 100000000000$? Probably the computer would display something like "overflow" on the screen. Why?

Clear

In Matlab there is a command "who" which specifies all your current variables. If you applied that now you would get the answer: Your variables are a, b, c, *ans*.

There is also a command "clear" which empties all boxes and removes all box names/labels (except ans). Apply that now!

Integers

On your computer keyboard there is a (strange) key marked, or labeled, $-$. What happens in the computer as a result of typing $a = -2$? Before we go into this you should read the rest of chapter 5 of *AMB&S*, about the extention of the natural numbers to *integers*.

So, what happens in your "match"-computer as you type $a = -2$. Well, this should be something that added to 2 gives 0, that is, some kind of "anti-matches". As a symbol of anti-matches we could use burned matches. So burn off two matches and put into the a-box. Now discribe in detail what happens it you type $a + b$. Recall that as an "anti-match" meets a match they both disappear. Finally, show what happens if you now type $c = c - b$ (recall that $c - b$ means $c + (-b)$) and then $a = b - a$.

Multiplication and powers

Multiplication is (must be!) denoted by * in Matlab. Thus, instead of writing $a + a + a + a$ one may write $4 * a$, or $a * 4$.

Exercise With $a = -2$ as above, what will the result be of writing $b = 3 * a$?

Exercise What is the result of $c = a * a$?

Like * provides a shorthand notation for *repeated addition*, the power ˆ provides shorthand notation for *repeated multiplication*. Thus, instead of writing $a * a * a * a$ we may write $a\hat{}4$ (but NOT $4\hat{}a$!). In normal hand writing we write this as a^4.

Exercise What is the result of writing $2\hat{}3$, $3\hat{}2$, $a\hat{}3$?

The following rules for implicit presedence are common standard. Powers are evaluated first, then multiplications and last additions. If you think of powers as being abreviations for multiplications, and multiplications for repeated addition, this is quite natural! Thus $2 + 3 * 4$ is 14 (not 20!). If you mean $5 * 4$ you have to write $(2 + 3) * 4$, using paranteses to override the presedence rules.

Exercise What is the value of $2 * (3 + 3\hat{}2)$?

Of course, the computer can work also with *fractionals* or *rational* numbers such as 3/2 and 8/11.

Exercise Demonstrate what happens in the computer as you type $a = 3/2$ (put one and a half (unburned) match into a), then $b = -0.5$ and finally $c = 3 * a - b - 1$.

Equations

Equations are important in mathematics. They model such things as *balance* and equilibria. The problems below may require that you also have read chapter 6 in *AMB*&S about the extension of the integer number system to the system of *rational* numbers.

Solve the equations (find x such that) a) $x + 3 = 7$ b) $3\,x = 12$ c) $3x - 1 = 19 - 2x$ d) $x/(6 - x) = 2$ The important thing here is not to find the actual solution value, but to fully understand the solution process, that is, being able to motivate each step in the process of finding x.

If above you could refer to a certain solution proceedure, now try to solve e) $x^3 = 64$ f) $2^x = 16$. Think of *how* you solved these equations. How would you discribe the solution process/algorithm. Discuss in the group how you found the answer. It was by some kind of "trial and error", wasn't it?

What is the chance of being able to solve a given equation

By "solve" we here mean systematically manipulate the equation to bring it to the form $x = ..$, displaying the solution. To get an idea we consider (relatively simple) equations consisting of two instances of the desired value x, three constants 3, 4 and 5, the three "operators" $+$, $*$ and ^ , and finally the $=$. Write down these nine symbols on small pieces of paper and put the first five of them in one box and the last four into another box. Then pick a (random) constant or x variable from the first box, then an "operator" from the second box, then a constant or x again from the first box, and so on until both boxes are empty, and put all the symbols in a line to make an equation like, for example, $3\hat{}x + 4 = 5 * x$. Then discuss/determine wheter or not you can solve the resulting equation. Repeat the procedure, say, ten times, to find an approximate probability for an equation of this form to be solvable in the given sense. Does the result surprise you?

Systems of equations

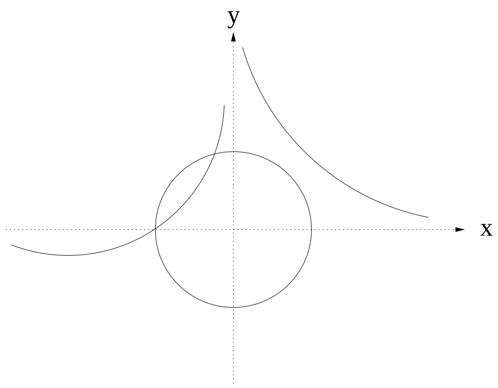
Many equations have the form of "systems" of two or more equations. Consider for example the two (coupled) equations

$$\begin{cases} x^2 + y^2 = 1 \\ x^2 y = x + 1, \end{cases} \tag{1}$$

and note that the idea is to find an x and a y that fits into *both* equations!

Make another version of the set of constants, variable and operators, but now denote the variables by y instead, and let one of the y's chance place with one of the x-es. Now pull sets of two equations from the two sets of symbols, each containing one x and one y, forming a system of two equations for the two unknowns x and y. What is the chance of being able to solve such a system, that is, to find an x and a y that fits into both equations, do you think? Discuss in the group. Dissapointing?

Exercise What has the following figure to do with the system of equations in the above example? Can you find a solution of the equations? Is there another solution? Can you find it?



Trial and error

A never failing method for equation solving is the "trial and error" method, where failing means getting stuck and not being able to continue the process. The method consists of picking an x, by *random* in the simplest case, and then try if it fits into the given equation or not. If it fits you have found a solution, and if it doesn't fit you can always pick another x for a new trial! Test this method on the equation $x^3 + x = 3$. For example, as a first "trial solution" you can pick $x = 1$. For this x you find at once that $x^3 + x = 2 < 3$. Perhaps x was too small. Lets try with $x = 2$ to find that $x^3 + x = 10 > 3$. So why not try $x = 1.2$ (why?, well, why not?). In any case you find that $x^3 + x = 2.92 < 3$. Hm, seems that we are getting close to something. If you patiently continue in this way, you are likely to get quite close to a solution after a while. What you should learn from this is that the trial and error method is a perfectly legal solution procedure, and in fact the only one available in most cases. What we need to add in the next course are some good strategies for picking the new trials from experience gained from the trials before. Of course, pure trial and error without any such strategy for getting closer to the solution is not likely to hit the target in a limited amount of time.

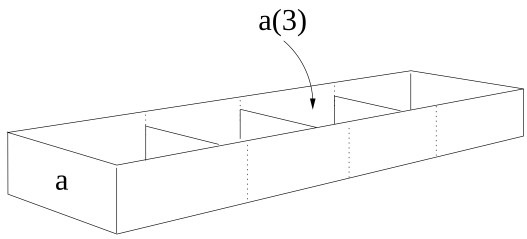
Problem solving

How many "sections" of paper is there on a toilett paper roll?

Array/list variables

An *array* or *list* variable is a variable that can hold several different values at the same time! One example could be the size of a (3-dimensional) box, characterized by length, width and height values. Instead of storing these values in 3 different variables (labeled boxLength, boxWidth and boxHeight, for example), we could store the 3 size variables of the box in boxSize(1), boxSize(2) and boxSize(3). We here view boxSize as *one* (list/array) variable, and we access the different values of the variable through their *position* in the list; where, for example, the first value refers to the length, the second to the width, and the third refers to the height. Another example could be the *date*. Instead of introducing year, month and day variables we could use a *date variable* with 3 values date(1), date(2) and date(3) for the year, month and day, respectively.

One can picture a list variable as in the figure below, where the array/list has the (simple) name a and can hold 4 values $a(1)$ through $a(4)$.



You can then address the complete list (of values) in the variable by a, or an individual value by, for example, $a(3)$.

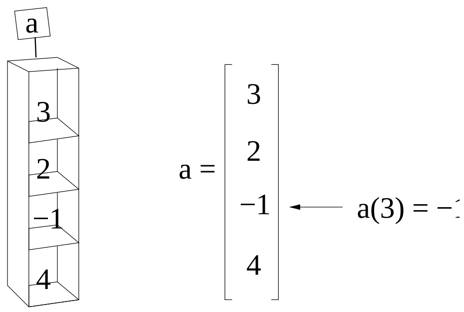
Exercise Build an array "box variable" suitable for storing the current date, label the variable "date", and fill its three sub-boxes with appropriate values (number of matches) corresponding to the current year (of the millenium), month and day, respectively.

Exercise Produce an array or list variable as the one in the figure and label the boxes a(1), a(2), a(3), a(4). Then demonstrate what happens in the computer as you type $a(3) = 2$, $a(1) = 3$ and $a(4) = a(1) + a(3)$.

One remark that can be made about list variables is that if you now type (ask for the value of) $a(2)$, which haven't been specified, you would find that it has been given a *default* value of 0. On the other hand, if you type (ask for the value of) $a(5)$ you will get the message that this value has not been defined. In fact, Matlab builds array variables dynamically. In the example, as you first type $a(3) = 2$, Matlabs builds an array consisting of 3 boxes, and fills the third of these with the value 2, and the two preceeding with (default) zero values. As you later define $a(4)$ Matlab glues another box to the three first ones and fills this with the value 5 (corresponding to the sum of $a(1)$ and $a(3)$). However, a fifth box is not created until you define $a(5)$ or $a(i)$ for some $i > 5$.

The advantage of using list variables becomes more appearent for data characterized by larger number of values. For example, if we would like to store the number of days in the different months of the year it would be natural to use a list variable daysInMonth with 12 values daysInMonth(1) through daysInMonth(12). To fill this list "variable" with appropriate values we write daysInMonth(1)=31 (for January), daysInMonth(2)=28 (for February), etc, or shorter daysInMonth=[31, 28, 31, .., 31], which is a shorthand notation for the same initialization.

Of course, one may also think of a list variable as a "vertical" set of boxes as in the following figure. Note also the corresponding symbolic notation to

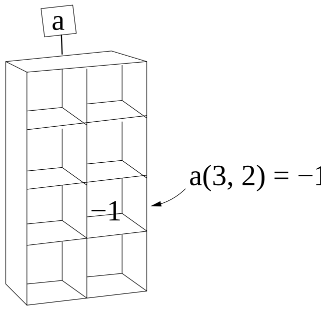


the right.

One may also consider *two-dimensional* array variables like in the figure below.

String variables

Another "type" of variable, very common in computers, have *string* values. A (text)*string* is just a sequence of letters or "characters", like 'hello' or 'ft45&ssR' (the latter of which could be a computer password for example).



Exercise Write hello on a piece of paper and put it into the box a, and model this by writing (on a piece of paper) a ='hello'.

Note that for example 2 to the computer could mean two different things; on one hand the *number* 2, and on the other hand the (one letter short) *text string* or *character* 2.

Exercise Visualize the difference of $a = 2$ and a ='2' by putting different things into the box a.

Strings are also arrays

In fact, *string* variables are also arrays, but arrays of *characters* rather than of numbers. That is, if you define a ='hello' and then type (ask for) $a(2)$ you get the second character 'e' in return.

num2str, str2num and eval

We have already mentioned that '2' and 2 means two diffenrt things. The first is the *character* 2 while the second is the *number* 2. There are functions in Matlab which can *convert* from the number type to the character/string/text type, and vice versa. For example num2str(2)='2' does num(ber)2(to)str(ing) conversion of the number 2 to the string/character '2'. Similarly str2num('2')=2 converts the string/character '2' to the number 2. Another function that tries to make mathematical (number) sense out of a character, string och expression is eval. For example '3/2' is to Matlab just a text/string/sequance of three characters 3, / and 2, without mathematical meaning or value. It could represent the third of February perhaps. The function eval evaluates the expression, if possible, seeking its mathematical value, if there is one possible. For example eval('3/2')=1.5.

Exercise Demonstrate with the paper boxes, text on pieces of paper, and matches the computer action as you type $a = 2$, $b =' a + 3'$, eval(b), $a =$num2str(ans).