

Lecture 2 Mathcad Basics

Operators

+ Addition, - Subtraction, * Multiplication, / Division, ^ Power
 () Specify evaluation order

Order of Operations

| | |
|---|---------------------------------|
| () ' ^ | highest level, first priority |
| * / | next priority level |
| + - | last operations to be performed |
| $y := 2 \quad x := 3 * y^2 \quad x = 12$ | |
| $y := 2 \quad x := (3*y)^2 \quad x = 36$ | |
| $y := 2 \quad x := 3 * y + 2 \quad x = 8$ | |
| $z := 3*6+6*2/4 \quad z = 21$ | |
| $x := 5^2/2 \quad x = 12.5$ | |

Homework Format and Template:

We'll each develop a simple template to use to start any new homework. The idea of a template is to lay-out the basic structure of what goes into a homework assignment, minus the actual input, calculations, and output necessary for a specific assignment.

To create template:

Simply start a new worksheet and type in what you want on the template. Then use Save As and pick the mathcad Template option.

To use template:

From mathcad, open the template through the File --> Open command and change the Files of Type box at the bottom to Mathcad Template. Before adding any information to the template, save it as a standard worksheet of desired name by using File --> Save As. Note that the Save as Type box at the bottom automatically is set to standard Mathcad Worksheet format. Choose a name and save. Now the original template is preserved for the next use, and you can begin work on your new assignment.

The basic structure and ingredients for every homework is given below. We'll base our template on this.

Title Page:

Info: Name, Course, Assignment Number and Title.

Summary/Observations: A brief summary (description) of the assignment. What did the assignment ask for? What did you learn? Did you discover anything easy/difficult with the assignment? It doesn't have to be long, just clear and concise.

Pseudocode: Pseudocode is a detailed description of the steps needed to solve the problem from beginning to end.

Work Page(s):

Standard Code: Any Mathcad statements needed in every worksheet. (e.g. ORIGIN=1)

Input Section: Scalars, vectors, matrices needed to begin calculations. Includes both permanent assignments and user prompted information as well as data file loading.

Calculation Section: Manipulations of the input information

Output Section: This should include all your results (vectors, graphs, etc.). Clearly label what each piece of output is.

Hand Verification: A sample problem by hand or other means independent of Mathcad. You need not verify every calculation, just samples to show your Mathcad results work.

Fundamental Program Structure

Labeling the program using comments

- program title
- student information
- program summary

executable statements

- program input (load data from external files, assignment statements, etc.)
- perform operations needed (sequential execution, loops, etc.)
- display program output (graphs, numbers, output files, etc.)
- intersperse comments to explain program

Sample Homework Template

Computer Methods in Civil Engineering

CGN 3421
Fall Semester, 2001

Kurt Gurley

HW#:

HW Title:

Problem Statement:

Observations:

Pseudocode:

page break

ORIGIN≡ 1

INPUT SECTION

CALCULATIONS SECTION

OUTPUT SECTION

Version 2000 and Version 2001

Mathcad releases new versions of their product at a fairly quick pace. As a result, you will probably come across more than one version, including: Mathcad 8, Mathcad 2000, Mathcad 2001 (the latest). From 2000 or 2001, you can use Save As to save your worksheet in an older format if desired. This is necessary, for example, if you bought 2001 for your personal machine. Both 200 Weil and the Circa labs use Mathcad 2000 (we're not convinced its worth the price to upgrade yet). Newer versions of mathcad CAN open worksheets made on older versions (2001 can open a 2000 worksheet), but older versions CAN'T open worksheets from newer versions (2000 can not open a 2001 worksheet). So if you create a worksheet at home (2001), and try to open it at school, you won't be successful. You can avoid this using the Save As older versions feature. That is, 2001 can save into 2000 format so you can create in 2001, and open in 2000. You can still open in 2001, but now you can also open in 2000.

Regions

A region is a spot in the worksheet where you enter text, equations or graphs.

Equation regions:

By default, when you click the cursor, mathcad assumes you are entering an equation into this new region. We've already seen how to assign numbers to variables, etc. Every number-manipulating type of expression is considered an equation region, including control structures (if statements, for loops, etc.)

Text regions:

You can also place the cursor and start typing your text. You may find that typing text doesn't always look like you would like it to, since Mathcad expects an equation by default, and can make some assumptions you didn't intend. To ensure that the region you are starting is text (for headings, instructions, comments, etc.), just pace the cursor, then type " to initiate a text region. Be sure to try out the Format menu to practice with changing fonts, styles, colors etc. to give your worksheet a professional look.

Graphics regions:

These regions can be initiated from the Insert menu, from the View -> toolbars -> graphs tab, or keyboard shortcuts (typing @ creates a 2-D graph). You must fill in the placeholders for the x and y axes to tell Mathcad what to plot.

Inserting Equation regions into text regions

When using text to explain the contents of a worksheet (talking through the equations so others can follow your work), it is possible to import an equation region directly into a text region. This can be done by initiating your text, then placing the cursor where you want the equation, and using Insert -> Math Region.

Enabled and Disabled Equation Regions

You may find occasion to wish you could ‘turn off’ an equation without actually deleting it. For example, an equation that appears for explanatory reasons within text may only be for show, not for function. Other reasons will come about as you begin to debug more complicated worksheets, and you want to see what happens if equation A wasn’t there (but you want to save it for later). Disabling an equation is easily done with the cursor by right-clicking in a equation region and clicking ‘Disable Evaluation’. A small mark will appear by the equation to let you know its disabled. It will remain disabled until you turn it back on in the same manner (right-click, and click Enable Evaluation).

Alignment of regions

After entering numerous different text, equation, and graphics regions, the look of the worksheet can get to look a little messy. For example, headings for sections may not be in line with each other vertically, equations that are next to each other may not be aligned exactly. You can adjust each region individually to fix this, but this can be imprecise, and tedious. Mathcad has an automated aligning feature to help neaten-up the look. Try entering three separate lines of text. Now use the mouse to highlight all three regions. Now use Format -> Align regions to line them all up neatly. If you have many regions, and desire different alignments to different groups of regions, you can hold the Ctrl button and left click selected regions, then align.

Areas

An area is a section of the worksheet you create, in which you can place various regions. The purpose is to be able to group regions that you might want to hide or even lock from other users seeing, changing, or both. All equation and graphics regions within an area work just like they would anywhere else. Try creating a region by placing the cursor and using Insert --> Area. You will see two horizontal lines span the page. The area is the region between the lines. You can enlarge the area by clicking the upper or lower line defining the area and dragging it up or down. You can now drag regions into the area, or create new regions within the area.

Collapsing areas

The reason for creating areas is to allow you to collapse or hide groups or regions without losing their function. After setting up an area and creating a few regions within it, double click the upper or lower area line to collapse the area (you’ll see the regions disappear). Another method is to left click the upper or lower area line and then right click to make options appear, including ‘collapse’.

locking areas

You can also lock an area so that you (or more important, someone else) can’t change any regions inside the area. The equations still work. No one can change anything in the region until they enter a password that you determine when you lock the area. Of course, you can combine the locking and collapsing so that no one can even look at the regions in the area until entering the password.

Range Variables and Vectors

A range variable is a variable that contains a series of numbers rather than a single number (but its not a vector). A range variable is different than a vector in that it cannot be referenced for specific values. All the values in a range variable must be used at the same time. If a range variable appears in an equation, it implies that the equation should be executed for each value in the range variable. We can also use range variables as another way to create vectors.

In general, vectors are much more flexible than range variables in terms of what you can do with them in Mathcad. We will learn to use range variables as a tool to create vectors, then use the vectors to carry out calculations and create graphs.

Creating Vectors using Range Variables

The general format to create a range variable is :

`i := start, next ; end`

this creates a range variable (not a vector) starting at the value 'start', and ending at the number 'end'. 'next' determines the increment by stating the next number after 'start', and the same increment between start the next is used to built the entire range variable. Therefore a range variable must consist of a number of equally spaced values. If you do not specify the 'next' value, Mathcad uses a default stepping value of 1.

Let's use a range variable to create a vector.

`i := 1..5`

create a range variable i

`veci := 1 + 0.5*i`

use the range variable as an index to a vector we are creating
use the range variable in the equation for the vector definition

`vec =`
$$\begin{pmatrix} 1.5 \\ 2 \\ 2.5 \\ 3 \\ 3.5 \end{pmatrix}$$

view the vector we just created

`vec3 = 2.5`

index a specific value inside vec

`i3 =` ■

cant view a specific value within a range vector

Notice: the range variable 'i' is used to specify the index to the vector 'vec' being created. This only works if the contents of the range variable are integers

Notice: the range variable 'i' also appears in the equation

Generalizing end points of an evenly spaced vector:

We can generalize the previous operation to arbitrary starting and ending points. We want to create a vector that spans from -2 to 5 in even steps of 10.

first enter scalars and calculate increment

```
begin := -2      end := 5      NP := 10      inc := (end-begin)/(NP-1)
```

Now create the range variable 'i' and use it to create the vector 'x'

```
i := 1;10      xi := begin + (i-1)*inc
```

inc = 0.778

| | | | | | | | | | | | |
|---------|---|----|--------|--------|-------|-------|-------|-------|-------|-------|----|
| $x^T =$ | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 1 | -2 | -1.222 | -0.444 | 0.333 | 1.111 | 1.889 | 2.667 | 3.444 | 4.222 | 5 |

Creating 2-D arrays using range variables:

We can extend the idea to arrays of 2 dimensions. Suppose I want a 3 by 5 matrix in which each element is the sum of its indices...

```
i := 1;3      j := 1;5      zi,j := i + j
```

$$z = \begin{pmatrix} 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 7 & 8 \end{pmatrix}$$

Note that the operation on the right hand side of the equal sign is calculated for every combination of the values in the range variables i and j.

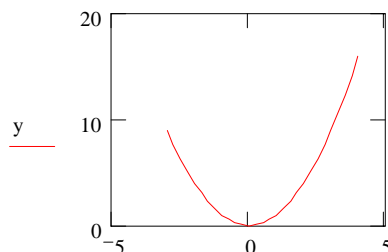
Generalizing to nonlinear relationships:

How can I create a vector that contains the expression: $y = x^2$ for $x = [-3, 4]$ in steps of 0.25 ??

We can create two vectors (x,y) and pull the same stuff.

```
begin := -3      end := 4      inc := 0.25      NP :=  $\frac{\text{end} - \text{begin}}{\text{inc}} + 1$ 
```

```
i := 1.. NP      xi := begin + (i - 1)·inc      yi := (xi)2
```



Functions: A first look

A single or multiple commands can be assigned to a single word (or letter) that represents those commands, followed by a list of inputs within parentheses. Once this assignment is made, you can re-issue all of those commands again and again by using the single word. A simple example is the creation of a long equation that you intend to use more than once. Why re-type over and over?

To the right I've created a function called 'y' which is a function of 't'. To use it, just 'type y(number)= ' to view the result. Note that we can send in scalar values or a range variable 'i:=1,1.5..2.5' which produces 4 answers.

$$y(t) := 5 + 0.6t - .2 \cdot t^2 + \log(\sin(t) + 2)$$

$$y(.1) = 5.38 \qquad y(30) = -156.995$$

$$x := 6 + y(4)^2 \qquad x = 24.443$$

$$i := 1, 1.5 .. 2.5 \qquad y(i) =$$

| |
|-------|
| 5.854 |
| 5.927 |
| 5.864 |
| 5.665 |

A function with more than one line of commands has a distinct look by the use of a vertical line to the right of the name we give the function, and to the left of the statements in the function. Let's say we want to create a function to solve the quadratic equation, and we break down the solution into parts. The way to create the vertical line is either the right square bracket, or go to 'add line' from the 'programming' toolbar (under view => toolbars)

$$\text{QuadEq}(a, b, c) := \left[\begin{array}{l} \text{radical} \leftarrow \sqrt{b^2 - 4 \cdot a \cdot c} \\ \text{ans}_1 \leftarrow \frac{-b + \text{radical}}{2 \cdot a} \\ \text{ans}_2 \leftarrow \frac{-b - \text{radical}}{2 \cdot a} \\ \text{ans} \end{array} \right.$$

QuadEq solves the quadratic equation of the form $a \cdot x^2 + b \cdot x + c = 0$

ans is the 2x1 vector to be reported as output from the function

$$\text{QuadEq}(2, 3, 1) = \begin{pmatrix} -0.5 \\ -1 \end{pmatrix} \qquad \text{QuadEq}(4, 0, -16) = \begin{pmatrix} 2 \\ -2 \end{pmatrix} \qquad \text{radical} :=$$

Note that the solution requires 3 pieces of input information, and the output has two pieces of information. We can now call this function from wherever we like in this worksheet (after the function definition). Of course I validated the answers I got to make sure they are correct.

IMPORTANT: None of the variables created in the function are recognized outside the function, except those listed at the end. In this case, 'ans' is listed at the end, while 'radical' is not. Therefore, the contents of 'ans' is transferred out to become the answers, while 'radical' is not recognized outside the function. You can see this above. I tried to type 'radical = ', but Mathcad did not recognize the variable name, and

is trying to force me to define it as a new variable name. 'ans' also does not exist outside the function, but its contents have been transferred as the result of 'QuadEq(2,3,1) = '

We will spend a good deal of time working with functions. This will be the main tool for developing our own programs. Once we learn the three basic control structures (decisions, counted loops, and conditional loops), the possibilities are almost endless with what we can do.

Another Example: Using a single-line function and a range variable to plot a curve

Plot the following function $y = 2 + 3x - 2x^2 + 0.2x^3$ over the range $x = [-5 \ 10]$

step 1) create the function 'y' as a function of an input variable 'x'

step 2) create a range variable 'x' from -5 to 10 in steps of 0.5

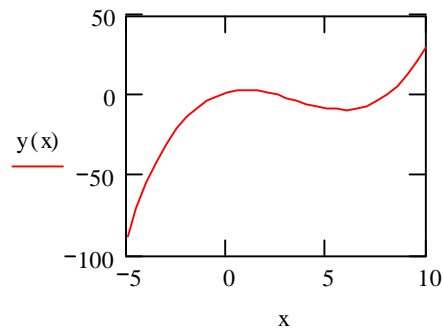
step 3) create a graph and place 'x' in the box along the x-axis, and 'y(x)' in the box along the y-axis

$$y(x) := 2 + 3x - 2x^2 + 0.2x^3$$

CREATE FUNCTION

$$x := -5, -4.5.. 10$$

CREATE Range Vector



PLOT of function