Understanding Ranges, Sequences, and Vectors

Introduction

New Mathcad users are sometimes confused by the difference between range variables and vectors. This is particularly true considering that other applications use expressions like 0..3 or 1:4 to define vectors.

It can be confusing when you type something like the expressions below and get an error message.

$k := 0, 0.2 .. 1$ a range variable definition

$v := \sin(k)$ a vector of sine values

Since Mathcad handles a vector as the argument to sin, you may expect that Mathcad interprets k as the vector $(0\ \ 0.2\ \ 0.4\ \ 0.6\ \ 0.6\ \ 0.8\ \ 1)^T$. If you evaluate sin(k) and k, you do get vector looking results.

$\sin(k) =$

0
0.199
0.389
0.565
0.717
0.841

$k =$

0
0.2
0.4
0.6
0.8
1

The aim of this article is to explain the difference between range variables and vectors, and why the expression v := sin(k) fails.

Just to show that there is light at the end of the tunnel, here is the correct way of using range variables in this context.

$k := 0, 1 .. 5$ convert range variable to use integers only

$t_k := 0.2 \cdot k$ for each value of k, set the kth element of vector variable, t, to 0.2 times the corresponding value of k

$v_k := \sin(t_k)$ for each value of k, apply sine to the kth element of vector t and assign the resulting value to the kth element of the vector stored in the variable v

Many Mathcad functions work on complete vectors, including sine. You can take advantage of this by simply writing:

$v := \sin(t)$ apply sine to t and assign the resulting vector to v

Note that:

- A vector is a one-column array and can store numbers.
- A vector can only have integer indices.
- A range definition, or 'range' for short, describes how to generate sequential numbers, but is not itself a list of numbers.
- A range variable is a variable that stores a range definition..
- Mathcad treats range variables as an implicit instruction to loop over each value in the implied sequence.

This probably still seems puzzling, so let's examine variables, vectors, and ranges in more detail.

What are Mathcad vectors and range variables?

Vectors

In Mathcad, a **vector** is a single column array of data. The elements of a vector, in common with its parent structure, the array, can hold scalars, strings, function names, or other arrays (forming a "nested" vector). For example, you can create a vector using the function stack and assign it to a variable:

$v := \mathrm{stack}(1,2)$

$w := \mathrm{stack}(1,\text{"a"},3)$

set x equal to w

$x := w$

add a vector to the end

$x_3 := v$

$$v = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \qquad w = \begin{pmatrix} 1 \\ \text{"a"} \\ 3 \end{pmatrix} \qquad x = \begin{bmatrix} 1 \\ \text{"a"} \\ 3 \\ \begin{pmatrix} 1 \\ 2 \end{pmatrix} \end{bmatrix}$$

There are some restrictions on the type of data that a vector can hold - for example, all elements must have the same units - but it is nevertheless a powerful tool for storing, accessing, and manipulating data. Indeed many Mathcad functions are explicitly designed to handle vectors.

You can extract the values of each element using the Mathcad array indexing notation v_n where n is an *integer* ranging from ORIGIN to last(v). Note that n must be an integer. Also note that a vector is a data type and not a variable; a variable in its turn can hold a vector, but isn't a vector itself.

Range Variables

A **range variable** is a variable that stores a range definition (or 'range' for short). When Mathcad encounters a range variable in an expression, it treats the range variable as a request to generate a set of numbers and to evaluate the expression for each one of those numbers in turn. In programming terms, a range variable is an implicit for loop. In the examples below, the range variable is on the left-hand side of the definition, and the corresponding range is on the right:

$$k := 0..4 \qquad t := 0, \frac{1}{4}\pi..\pi$$

$k =$

0
1
2
3
4

$t =$

0
0.785
1.571
2.356
3.142

Both a range and a vector are data types. The key to the differences between a vector and a range is that a vector is meant for storing data, while a range is shorthand for a list of values. You can readily access each element of a vector, but you *cannot* get at the individual elements of a range variable (indeed, it doesn't make sense to try). They don't exist, and Mathcad generates an error message if you do try.

$k_2 =$

To understand range variables a little bit better, let's look in detail at ranges and the related concept of a sequence, starting with the sequence.

Sequences

A sequence is a comma-separated list of values, where a value in this instance is a scalar, string, array, sequence, range, function name or an expression that evaluates to one of these types of value (for example an evaluated function or a range variable).

The following are valid sequences:

$0, 0.1, 0.2 \qquad 1, \text{"a"}, \text{"b"}, \text{"c"}, 2$ (the second sequence is valid in Mathcad 13 and 14, but not in Mathcad 11)

Note that a sequence is not a vector, and it isn't possible to pick out a single element from within it. A sequence also differs from a range variable in that it can't be assigned to a variable, which makes statements like the one below flag up an error if it is used.

$s := 0, 0.1, 0.2, 3$

$2s =$

Sequences are commonly used as arguments to functions or in for loops, which will be discussed in a future article.

Now let's turn back to the range.

Ranges

A range is an abbreviated specification for a uniformly spaced real numeric sequence. The syntax for a range is start.. end where end is a scalar and start is either a scalar or a 2-element sequence. A range can define either a set of increasing values or a set of decreasing values.

For example, 0..9 is an ascending range that starts with a scalar, while 0, -0.1..-1 is a descending range that starts with a sequence.

When Mathcad evaluates a range beginning with a sequence, it interprets the second number in the sequence as the *next* number in the sequence.

Note that the second value in the expression for the range is *not* the step size. The step size is the difference between the first two values. If the range starts with a scalar, then Mathcad assumes a step size of 1 in the appropriate direction. If the step is known, then the range can be readily defined as start, start + step.. end. For example,.

$$1 , 1 + 0.25 .. 2$$

A range can also form part of a sequence, so 1, 2, (0, 0.25 .. 1) is a valid sequence. Note that the parentheses are mandatory to delimit the range from the rest of the sequence. However, because Mathcad restricts a range to real scalars, "a" .."c" is not a valid range (these are strings not scalars).

Using Range Variables

Range variables have a number of uses within Mathcad, from quickly evaluating an expression over a range, to creating recursive sequences.

Show and Tell

The simplest use of range variables is in calculating an expression for a given list of numbers and displaying the results. The syntax is straightforward "expression =," where the expression contains one or more range variables. When Mathcad encounters this form, it evaluates the expression for each combination of values covered by the range variables. Mathcad then displays the results in table form.

In the example below, there are two range variables, i and j, that hold the ranges 4 .. 5 and 1 .. 2 respectively. Evaluating the expressions i^2 = and j^2 = causes Mathcad to calculate the squares of 4 then 5 for i, and 1 then 2 for j.

$$i := 4 .. 5 \qquad j := 1 .. 2 \qquad i^2 = \begin{array}{|r|}\hline 16 \\ \hline 25 \\ \hline \end{array} \qquad j^2 = \begin{array}{|r|}\hline 1 \\ \hline 4 \\ \hline \end{array}$$

In the combined expression $i^2 + j^2$, Mathcad fixes the value of j (the second range variable) and iterates over i, then Mathcad takes the next value of j and iterates over i again. Reversing the order of declaration reverses the order of evaluation.

$i^2 + j^2 =$

17	$4^2 + 1^2$
26	$5^2 + 1^2$
20	$4^2 + 2^2$
29	$5^2 + 2^2$

$j^2 + i^2 =$

17
20
26
29

You can use any type of range variable in this way, whether it has integer steps or fractional ones.

The sin(k) example at the beginning of this article has fractional steps. Remember that k isn't a vector - it's a range variable. When you evaluate sin(k), Mathcad iterates through k and displays a result for each value of k. The results appear in a grid to show they are related, but that grid does not represent an array - merely a visual representation of a list of individual results.

Pointing the Way

Evaluating an expression as described above is often what's needed. However, there are also many occasions where you may want to make use of these results later on in a worksheet.

Fortunately, Mathcad provides a convenient way to do this by allowing a range variable to serve as an array index. Consider the following pair of expressions:

$k := 0 .. 3$ range variable definition

$v_k := k^3$

- The first expression is the familiar range variable definition.
- The right-hand side of the second expression, k^3 contains the range variable and calculates the cube of each value in the range. If you write k^3 = you would see a table of cubes.
- The left-hand side of the second expression, v_k stores the values of k^3. It stores the result of evaluating k^3 in the kth element of the vector v.

$v^T = \begin{pmatrix} 0 & 1 & 8 & 27 \end{pmatrix}$ vector creation for storing values of k^3

Evaluating v^T reveals that you have created a vector, because the transpose operator T only works on arrays and generates an error if you apply it to a range variable.

$k^T =$

This mechanism extends to matrices as well; by using the previous definitions for i and j, you get

$A_{i-4,j} := i^2 + j^2$ $\quad$ $A = \begin{pmatrix} 0 & 17 & 20 \\ 0 & 26 & 29 \end{pmatrix}$

Here's another point about using range variables that helps distinguish them from vectors.

The only thing you can do to a range variable is define one to be equal to another one; Mathcad does not allow even basic arithmetic operations on a range variable.

$n := i$

$n := i + 1$

You can't add 1 to an existing range variable because the range is not a numeric type, but a specification for how to produce a set of numbers. It makes as little sense to add 1 to a range as it does to add 1 to a string.

The definition of A above has the expression i-4 in the row index so you may wonder, why Mathcads allow subtraction on the range variable i?. The answer is that Mathcad treats a range variable in an expression as an instruction to evaluate that expression for each value that the range specifies. In the definition above, i does not refer to the whole range variable, but to the values that it generates. Hence, the index expression i-4 is an instruction to Mathcad to take the "current" value of i and subtract 4 from it.

Now that you can calculate an array index, you can generate results that depend upon previously calculated values. A good example of this is the Fibonacci sequence (where "sequence" is used in its general mathematical sense).

The Fibonacci sequence is "1,1,2,3,5,8,13,21 ...", where each number in the sequence is the sum of the preceding two elements. To start the sequence, define the first two elements to both be 1. For the nth number, the indices of the two preceding number are simply n-1 and n-2. This is all you need to generate a vector holding the Fibonacci sequence.

$fib_0 := 1$ first number in the sequence

$fib_1 := 1$ second number in the sequence

$n := 2..7$ range variable to create the next 6 numbers

$fib_n := fib_{n-1} + fib_{n-2}$ make the nth number equal to the sum of the preceding two numbers

$fib^T = \begin{pmatrix} 1 & 1 & 2 & 3 & 5 & 8 & 13 & 21 \end{pmatrix}$ show the resulting vector (transposed to make it easier to read)

If you hadn't defined the first two numbers in the sequence, Mathcad would either have automatically set them to zero (Mathcad 11) or flagged the error, invalid index (Mathcad 13 and Mathcad 14).

Here are are two more examples of common errors involving array indices. The first occurs when you write

$r := 0, 0.5 .. 1$

$w_r := r$

This case fails because an array index must be an integer, whereas r takes on fractional values. This second case looks as though it should work, but fails for a different reason.

$i := \text{stack}(0, 1, 2, 3, 4)$ create a vector of integers

$w_i := i$ try to use them to as indices into another vector

This case fails because Mathcad only knows how to handle array indices that are integers, which a vector isn't, and does not iterate over vectors, since they're just a *normal* data type.

Finally, there is one important piece of syntax to remember. When defining a variable in terms of a range variable, the range variable *must* appear on the left-hand side of the definition. Mathcad raises an error if it sees a range variable on the right, but not on the left.

Looking back at the original example,

$k := 0, 0.2 .. 1$

$v := \sin(k)$

you can see that k is a range variable and not a vector. You can also see that although k appears on the right of the second expression, it doesn't appear on the left, so Mathcad returns an error.

Naturally, for every rule there is an exception, and there are two cases where you can use a range variable on just the right of a definition - taking a range variable sum or product.

$\text{sumN} := \sum_{n} n \quad \text{sumN} = 27 \qquad \text{prodN} := \prod_{n} n \quad \text{prodN} = 5040$

Conclusion

In summary, a vector and a range are quite distinct entities. A vector stores values while a range specifies values but doesn't store them.

A range variable is a variable that holds a range. A range variable is an instruction to expand the range into a complete set of values and then evaluate each of those values in turn.

Understanding the differences between a range variable and other variables, particularly those holding vectors, is essential to knowing how to use them and when to use them.

The next issue of PTC Express will have a follow-up article on this subject by Stuart Bruff.

About the Author

Stuart Bruff has 25 years experience in engineering, both as a Royal Air Force communications engineer officer and subsequently as a systems engineer in the United Kingdom aerospace industry. He has used Mathcad since version 7 and is a regular contributor to the **Mathcad User's Forums**. He also works as a Mathcad consultant.