1. **Building Vectors and Arrays**

|  |  |
| --- | --- |
| **Array** | An n x m sized matrix containing n x m elements |
| **Vector** | Any one-dimensional array ie, 1 x n (row vector) or n x 1 (column vector) |
| **Double** | A data type that **holds numbers** in each of its elements |
| **Cell array** | A data type that can **hold different types of data** in each of its elements, including numbers, strings, or even other arrays of varying sizes. A cell array element can also hold an empty cell. |

There are many different ways to build the same **row vectors.**

|  |  |  |
| --- | --- | --- |
| [1 2 3 4 5 6 7 8 9] or  [1,2,3,4,5,6,7,8,9] | You can **individually type out each element,** separated by either a space or a comma (useful if you have a small, irregular data set) | |

To create a row vector containing **evenly spaced points** within a certain domain, you can use the built-in MATLAB function linspace() or the colon operator.

|  |  |  |
| --- | --- | --- |
| linspace(1,9,9)  ↑ ↑ ↑  **x1 x2 n** | **x1, x2**: domain limits  **n**: number of points you want between your limits; If you don’t specify **n**, it defaults to (x2-x1)/(n-1). | When you want to **specify the number of points** between your domain limits |
| 1:1:9 or 1:9  ↑ ↑ ↑  **x1 dx x2** | **x1, x2**: domain limits  **dx**: number of points you want to make within your limits; If you don’t specify dx, it defaults to 1. | When you want to **specify the size of the interval** between your domain limits |

|  |  |
| --- | --- |
| **Ex 1.** Create a vector called t representing a time domain from t = 0 to t = 30 days, with 2 day intervals:  **t = 0:2:30** | **Ex 2.** UV-Vis absorbance data is typically plotted against wavelength from 400 to 900 nm. You are given an evenly spaced dataset with 9235 data points in this range. How would you create the wavelength vector to plot this?  **wavelength = linspace(400,900,9235)** |

There are also multiple ways to build a **column vector**.

|  |  |
| --- | --- |
| [1; 2; 3; 4; 5; 6; 7; 8; 9] | Use a semicolon to separate rows in an array. |
| If A = 1:9, then A’ is the transpose (“column version”) of A | Transpose an array (ie, turn rows to columns) and columns to rows) |

You can use vectors to build **arrays**.

|  |  |  |
| --- | --- | --- |
| **Ex 3:** Find 2 different ways tobuild a 5x5 array called A whose columns are the integers 1 through 5, in order. | | |
| **A = [1:5; 1:5; 1:5; 1:5; 1:5]’** | **A = [ones(1,5); 2\*ones(1,5); 3\*ones(1,5); 4\*ones(1,5); 5\*ones(1,5)]** | |

1. **Indexing Arrays (2D)**

You will often need to change or access values that you have stored in your arrays. Sometimes you may need to build an array whose elements you will fill with data later. To access an element in an array, you use its **index**, **which represents the position of the data in the array.**

A complete 2D array index has two numbers: array(row,column).

For a row number **n** and a column number **m**:

|  |  |
| --- | --- |
| To access data in an array | value = array(n,m) |
| To change data in an array | array(n,m)= some value |
| To access an entire row or column of an array | values = array(n,:) or array(:,m) |
| To change an entire row or column of an array | array(n,:) = values or array(:,m) = values |

Consider a 3x7 array containing temperature readings for 3 different times across the 7 days of the week.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T =  31 37 35 34 31 29 32  44 46 47 45 39 39 42  40 42 31 44 33 38 37 | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  | **S** | **M** | **T** | **W** | **Th** | **F** | **S** | | **9:00** | **31** | **37** | **35** | **34** | **31** | **29** | **32** | | **12:00** | **44** | **46** | **47** | **45** | **39** | **39** | **42** | | **18:00** | **40** | **42** | **31** | **44** | **33** | **38** | **37** | |

|  |  |  |
| --- | --- | --- |
| **Ex 1.** What is T(3,4)? What is T(:,4)? What is T(3,:)? What are the indices of when the temperature is 44 degrees (in green)? | **Ex 2.** Data was incorrectly recorded for Tuesday (in blue) at 18:00. How would I change that entry to the correct value, 41 degrees? | **Ex 3.** All the data taken at 9:00 (in red) needs to be corrected for a calibration error that made the reading 20% lower than the actual temperature. How could I replace the old data with the corrected data? |
| **T(3,4) → 44**  **T(:,4) → [34;45;44]**  **T(3,:) = [40 42 31 44 33 38 37]** | **T(3,3) = 41** | **T(1,:) = T(1,:)\*1.2** |

1. **Preallocating Arrays**

Sometimes you may need to create an array first and then insert data into a certain location as that data is calculated. In this case, you can **preallocate an array, ie, create an array of a certain size filled with dummy values that you replace with your own data.** There are some built-in MATLAB functions useful for preallocation:

|  |  |  |
| --- | --- | --- |
| zeros(n,m) | ones(n,m) | nan(n,m) |
| Creates an nxm array filled with 0s | Creates an nxm array filled with 1s | Creates an nxm array filled with nans, or “not a number”s. |
| zeros(2,3) =  0 0 0  0 0 0 | ones(2,3) =  1 1 1  1 1 1 | nan(2,3) =  NaN NaN NaN  NaN NaN NaN |

1. **Array Operations**

MATLAB allows you to carry out **standard matrix operations**, such as multiplication[[1]](#footnote-0), addition/subtraction, and finding the inverse of a matrix (hence MATrix LABoratory!).[[2]](#footnote-1)

|  |  |  |
| --- | --- | --- |
| A =  1 3 5  6 9 4  2 8 7 | B =  1 2  3 1  2 0 | C =  0 1 3  4 1 4  0 2 2 |
| A\*B =  20 5  41 21  40 12 | inv(A) =  0.39 0.24 -0.42  -0.43 -0.04 0.33  0.38 -0.03 -0.14 | A+C =  1 4 8  10 10 8  2 10 9 |

But what if I actually wanted to carry out an operation element by element? MATLAB also allows you to carry out **element-wise operations**, including multiplication, division, and exponentiation using the dot operator, indicated by a period followed by the operation: for example, element-wise multiplication would be ‘.\*’.

|  |  |  |
| --- | --- | --- |
| A.\*C =  0 3 15  24 9 16  0 16 14 | A./C =  Inf 3.00 1.67  1.50 9.00 1.00  Inf 4.00 3.50 | A.^2 =  1 9 25  36 81 16  4 64 49 |

Consider a 3x8 array C\_0 containing surface measurements of an initial contaminant concentration data (in μM) for a lake at t = 0. The vector t represents days after t = 0 at which you would like to calculate measurements.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C\_0 =   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | 7 | 7 | 6 | 5.5 | 5 | 4 | 3 | 2 | | 9 | 8.5 | 8 | 8 | 7.5 | 6 | 5 | 3 | | 6 | 6 | 5.5 | 5 | 5 | 4 | 2 | 1.5 |   t =  0 3 6 9 12 15 18 21 24 27 30 | Lake Length   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  | **1 km** | **2 km** | **3 km** | **4 km** | **5 km** | **6 km** | **7 km** | | **2 km** | 7 | 7 | 6 | 5.5 | 5 | 4 | 3 | | **4 km** | 9 | 8.5 | 8 | 8 | 7.5 | 6 | 5 | | **6 km** | 6 | 6 | 5.5 | 5 | 5 | 4 | 2 | |

Suppose the contaminant decays according to a first order reaction such that the concentration at a given time is given by the equation:

C(t) = C0e-k\*t

where k is a rate constant equal to 0.1 day-1, C0 is the initial concentration such that C(0) = C0 (represented at each point in the lake by the array C\_0 above), and t is the time that has passed, in days.

|  |  |  |
| --- | --- | --- |
| **Ex 1.** How would you calculate an array that represents the contaminant concentration in the lake when t = 24 days? Create an array called C\_24d with these values. | **Ex 2.** How would you calculate an array that represents the contaminant concentration across the middle of the lake (4 km) at t = 24 days? Create an array called C\_24d\_4km with these values. | **Ex 3**. You would now like to see how the concentration across the middle of the lake (4 km) changes every 3 days from 0 to 30 days. Calculate an array called C\_4km with these values. (Hint: your resulting array will be 3x11.) |
| **C\_24 = t(9).\*C\_0** | **C\_24\_4km = t(9).\*C\_0(:,4)** | **C\_4km = t.\*C\_0** |

1. Recall that you can only multiply two matrices together if their dimensions follow the form N×M • M×P. The resulting matrix after multiplication will have dimensions N×P. [↑](#footnote-ref-0)
2. Note that the inverse of a matrix A, A-1, is defined as **A • A-1 = I**, where **I** is the identity matrix. The inverse of a matrix is NOT the reverse of matrix multiplication, and there is no such thing as matrix division. [↑](#footnote-ref-1)