**Arrays, Matrices, Vectors**

- **Array** is a collection of indexed data. For example, seats in a movie theater can be uniquely identified by two indexes: row number and seat number.
- Arrays can be one-dimensional (one index), two-dimensional (2 indexes), etc.
- **Vector** is a one-dimensional array of data. The vector elements can be labeled as $v_i$ where $i = 0, 1, 2, \ldots$ There are column-vectors and row-vectors:

$$V(\text{column vector}) = \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

$$H(\text{row vector}) = \left( h_0 \ h_1 \ h_2 \ h_3 \right)$$

- **Matrix** is a two-dimensional array of numbers arranged in a table format. The matrix elements can be labeled $a_{ij}$, where $i$ is the row index and $j$ is the column index.

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix}$$

- **Linear algebra** is the area of mathematics that covers manipulations of matrices and vectors. Chem 5 students will just touch on the basics. More ⇒ Math 2J
Matrices and Vectors in Mathcad

Matrix support in Mathcad
- **Vector** = 1D array of numbers
- **Matrix** = 2D array of numbers
- More than 2 array dimensions are not possible in Mathcad (limitation)

Define a 3-component vector and display its elements
\[ v := \begin{pmatrix} 10 \\ 20 \\ 30 \end{pmatrix} \]
\[ v_0 = 10 \quad v_1 = 20 \quad v_2 = 30 \]

Index subscript \( v[0] \)

The index of the first array's element is 0 by default, but it can be changed to any number by changing the predefined variable "ORIGIN"

\[ \text{ORIGIN} := 1 \]

\[ M := \begin{pmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \end{pmatrix} \]

Redefine array origin from 0 to 1.
Define a 3 x 4 matrix and display its selected elements
\[ M_{2,1} = 21 \quad M_{3,4} = 34 \]
Inserting Arrays

1. Create a matrix placeholder and fill it in

2. Initialize the matrix by giving a value to its bottom-right element, and then fill the rest in by defining each element.

   \[
   \begin{bmatrix}
   0 & 0 & 0 \\
   0 & 0 & 0 \\
   0 & 0 & 3
   \end{bmatrix}
   \]

   Initialize a 3 x 3 matrix. The elements that have not been defined yet are =0

   \[
   \begin{bmatrix}
   1 & 0 & 0 \\
   0 & 2 & 0 \\
   0 & 0 & 3
   \end{bmatrix}
   \]

   Define more matrix elements

3. Read matrix values from a text file using Insert → Data → File Input

   File test.txt containing tab-delimited text:

   1  2  3  4  5
   6  7  8  9 10
   11 12 13 14 15

   \[
   M = \begin{bmatrix}
   1 & 2 & 3 & 4 & 5 \\
   6 & 7 & 8 & 9 & 10 \\
   11 & 12 & 13 & 14 & 15
   \end{bmatrix}
   \]
Inserting Arrays

4. Using **Insert → Data → Table** menu and filling in the table

\[
L := \begin{pmatrix}
0 & 0 \\
0 & 1 \\
1 & 1 \\
2 & 2 \\
3 & 4 \\
4 & 9 \\
\end{pmatrix}
\]

\[
L = \begin{pmatrix}
0 & 0 \\
1 & 1 \\
2 & 4 \\
3 & 9 \\
\end{pmatrix}
\]

5. Copy/Paste data from a different application, e.g., Excel

<table>
<thead>
<tr>
<th>number</th>
<th>Formula =A2^2</th>
<th>Formula =A2^3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>125</td>
</tr>
<tr>
<td>6</td>
<td>36</td>
<td>216</td>
</tr>
</tbody>
</table>

Select the numbers and paste them in the placeholder. Mathcad will automatically convert Excel formulas into numbers.

\[
W := \begin{pmatrix}
1 & 1 & 1 \\
2 & 4 & 8 \\
3 & 9 & 27 \\
4 & 16 & 64 \\
5 & 25 & 125 \\
6 & 36 & 216 \\
\end{pmatrix}
\]
Range Variables

- Range variables take a series of values. Calculations where these variables appear are repeated for each of these values.
  - \( i:=1..20 \) assigns range variable \( i \) to values 1, 2, 3 ... 20
  - \( i:=20..1 \) assigns \( i \) to values 20, 19, 18 ... 1
  - \( i:=5,3..-7 \) assigns \( i \) to values 5, 3, 1, -1 ... -7
- They are entered by typing \([\text{name}][:]\text{[initial value]}[;][\text{final value}]\)
- Mathcad converts \([;]\) into \([..]\), and \([:]\) into \([:=]\) for the display purposes
- Once the range variables are defined, one can use them to fill an array:

\[
\begin{align*}
  i &:= 0..2 \\
  j &:= 0..3
\end{align*}
\]

\[
\text{Define two range variables}
\]

\[
\text{Vector}_{i} := i^2 \quad \text{Vector} = \begin{pmatrix} 0 \\ 1 \\ 4 \end{pmatrix}
\]

\[
\text{Calculate vector elements using range variable } i
\]

\[
\text{Mat}_{i,j} := i + j \quad \text{Mat} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \end{pmatrix}
\]

\[
\text{Calculate matrix elements using range variables } i \text{ and } j
\]

* Students are often confused about the way range variables work. Whenever a range variable is used in a Mathcad expression, the calculation is repeated for EACH value of the range variable. In the Vector definition example, the calculation is done 3 times, first with \( i=0 \), then with \( i=1 \), then with \( i=2 \). In the Mat definition example, the calculation is done 12 times with each possible combination of \( i=0,1,2 \) and \( j=0,1,2,3 \).
Example of Using Range Variables

In this example, we are generating a linear plot with a superimposed random noise. You can control the slope of the line, the intercept, the magnitude of the noise, and the number of points in your vectors. Play with all these values to see how the plot responds to them.

**Reset the array origin at i=1**

```
ORIGIN := 1
```

**Define a range variable**

```
i := 1..100
```

**Define constants**

```
Noise := 10   Slope := 1   Intercept := 10
```

**Define vectors**

```
RR_i := 2Noise·(md(1) − 0.5)   LL_i := Slope·i + Intercept
```

**Find the sum:**

```
SS := RR + LL
```

![Graph showing linear plot with random noise](image-url)
Using matrix() Function

Function \textit{matrix}(\texttt{Nrows, Ncols, f}) uses a different approach to define elements of a matrix, where \texttt{Nrows} = number of rows, \texttt{Ncols} = number of columns. A two-variable function \texttt{f(rr, cc)} is used to specify the calculation, where \texttt{rr} and \texttt{cc} are two dummy variables corresponding to the row index and column index. Note that \texttt{rr} and \texttt{cc} are NOT range variables, and that the order of \texttt{rr} and \texttt{cc} in the function definition is important.

\begin{align*}
\text{f1}(rr, cc) & := rr + cc^2 \\
\text{f2}(rr, cc) & := rr^2 + cc
\end{align*}

\[
M := \text{matrix}(4, 4, \texttt{f1}) = \begin{pmatrix}
0 & 1 & 4 & 9 \\
1 & 2 & 5 & 10 \\
2 & 3 & 6 & 11 \\
3 & 4 & 7 & 12
\end{pmatrix}
\]

\[
M := \text{matrix}(4, 4, \texttt{f2}) = \begin{pmatrix}
0 & 1 & 2 & 3 \\
1 & 2 & 3 & 4 \\
4 & 5 & 6 & 7 \\
9 & 10 & 11 & 12
\end{pmatrix}
\]

The calculation is repeated until all values in the requested matrix are calculated for all the rows and columns (\texttt{Nrow, Ncols}) specified in \textit{matrix()} function.
Things To Watch Out For

If you define a matrix or vector in your worksheet and then forget about it and try to redefine it in a different dimensionality you may get unexpected results:

Here we define \( S \) as a 3 x 3 matrix

\[
S := \begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\]

\( i := 0..2 \)

Here we try to redefine \( S \) a vector. But instead we are simply modifying one column of the previously defined matrix \( S \)!

\[
S := 7
\]

\[
S = \begin{pmatrix}
7 & 0 & 0 \\
7 & 0 & 0 \\
7 & 0 & 1 \\
\end{pmatrix}
\]

\( i := 0..3 \)

If we specify a larger index than the current size of matrix \( S \), it will automatically expand to the minimal size required to support that index.

\[
S := 8
\]

\[
S = \begin{pmatrix}
8 & 0 & 0 \\
8 & 0 & 0 \\
8 & 0 & 1 \\
8 & 0 & 0 \\
\end{pmatrix}
\]

\[
\text{\( S \), \( i := 3 \)}
\]

\[
S = \begin{pmatrix}
8 & 0 & 0 & 0 \\
3 & 3 & 3 & 3 \\
8 & 0 & 1 & 0 \\
8 & 0 & 0 & 0 \\
\end{pmatrix}
\]
Practice

Using range variables, build matrices with the following elements:

\[
A = \begin{bmatrix}
0 & -1 & -2 & -3 & -4 & -5 \\
1 & 0 & -1 & -2 & -3 & -4 \\
2 & 1 & 0 & -1 & -2 & -3 \\
3 & 2 & 1 & 0 & -1 & -2 \\
4 & 3 & 2 & 1 & 0 & -1 \\
5 & 4 & 3 & 2 & 1 & 0
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
0 & 1 & 2 & 3 & 4 & 5 \\
1 & 2 & 3 & 4 & 5 & 6 \\
2 & 3 & 4 & 5 & 6 & 7 \\
3 & 4 & 5 & 6 & 7 & 8 \\
4 & 5 & 6 & 7 & 8 & 9 \\
5 & 6 & 7 & 8 & 9 & 10
\end{bmatrix}
\]

\[
C = \begin{bmatrix}
0 & 1 & 2 & 3 & 4 & 5 \\
1 & 0 & 3 & 4 & 5 & 6 \\
2 & 3 & 0 & 5 & 6 & 7 \\
3 & 4 & 5 & 0 & 7 & 8 \\
4 & 5 & 6 & 7 & 0 & 9 \\
5 & 6 & 7 & 8 & 9 & 0
\end{bmatrix}
\]

4.1 Matrix Operations

Given the matrices and vectors

\[
I = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\]

\[
a = \begin{bmatrix}
1 \\
2 \\
3 \\
4
\end{bmatrix},
\]

\[
b = \begin{bmatrix}
2 \\
4 \\
6 \\
8
\end{bmatrix},
\]

\[
c = \begin{bmatrix}
2 & 3 & 7 & 11 \\
1 & 4 & 3 & 9 \\
0 & 6 & 5 & 1 \\
1 & 8 & 4 & 2
\end{bmatrix},
\]

which of the following matrix operations are allowed?

(a) \(I^T\) transpose the identity matrix

(b) \(|a|\) find the determinant of vector \(a\)

(c) \(a^{-1}\) invert the \(a\) vector

(d) \(|C|\) find the determinant of matrix \(C\)

(e) \(C^{-1}\) invert the \(C\) matrix

(f) \(I \cdot a\) multiply the identity matrix by vector \(a\)

(g) \(a \cdot b\) multiply the \(a\) vector by vector \(b\)

(h) \(b \cdot a\) multiply the \(b\) vector by vector \(a\)

(i) \(C^{-1} \cdot a\) multiply the inverse of the \(C\) matrix by vector \(a\)

For each operation that can be performed, what is the result?

4.3 Simultaneous Equations, II

Write the following sets of simultan (if possible):

(a) \(3x_1 + 1x_2 + 5x_3 = 20\)

\(2x_1 + 3x_2 - 1x_3 = 5\)

\(-1x_1 + 4x_2 = 7\)

(b) \(6x_1 + 2x_2 + 8x_3 = 14\)

\(x_1 + 3x_2 + 4x_3 = 5\)

\(5x_1 + 6x_2 + 2x_3 = 7\)

(c) \(4y_1 + 2y_2 + 1y_3 + 5y_4 = 52.9\)

\(3y_1 + y_2 + 4y_3 + 7y_4 = 74.2\)

\(2y_1 + 3y_2 + y_3 + 6y_4 = 58.3\)

\(3y_1 + y_2 + y_3 + 3y_4 = 34.2\)

Problems are taken from the course textbook:
Ronald Larson, Introduction to Mathcad 13
Matrix Operations

Possible matrix operations include the following:

- Addition/subtraction of matrices of the same size.
- Multiplication of two matrices. Size restrictions apply! According to the rules of matrix multiplication, a product of $n \times m$ matrix and $m \times k$ matrix will have the size of $n \times k$.
- Transposing the matrix (exchanging the row and column indexes)
- Inverting the matrix. Can be applied only to non-singular square matrix.
- Calculating determinant of a matrix. Can be applied only to square matrices.
- Sorting the values of a vector (or in a column of a matrix) in ascending/descending order

Two range variables defined

\[ i := 0..2 \quad j := 0..2 \]

Two diagonal 3 x 3 matrices defined

\[ A_{i, i} := i + 1 \quad B_{j, j} := 3 - j \]

Off diagonal elements manually corrected

\[ A_{0,2} := 1 \quad A_{0,2} := 1 \]

Sum := $A + B$ \quad Product := $A \cdot B$ \quad Inv$_A$ := $A^{-1}$

\[
A = \begin{pmatrix} 1 & 0 & 1 \\
0 & 2 & 0 \\
1 & 0 & 3 \end{pmatrix} \quad B = \begin{pmatrix} 3 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 1 \end{pmatrix} \quad Sum = \begin{pmatrix} 4 & 0 & 1 \\
0 & 4 & 0 \\
1 & 0 & 4 \end{pmatrix} \quad Inv_A = \begin{pmatrix} 1.5 & 0 & -0.5 \\
0 & 0.5 & 0 \\
-0.5 & 0 & 0.5 \end{pmatrix} \quad Product = \begin{pmatrix} 3 & 0 & 1 \\
0 & 4 & 0 \\
3 & 0 & 3 \end{pmatrix}
\]

Det$_A$ := $|A|$ \quad Det$_A$ = 4

Calculations
Vector Transformations

Multiplication of a square $n \times n$ matrix by a column vector of length $n$ always produces another column vector of length $n$. Therefore, this multiplication can be viewed as an operator that simultaneously rotates and stretches a vector in an $n$-dimensional space. To illustrate this example, here is what happens in a two-dimensional space (xy-plane):

Vectors $X$ and $Y$ can be represented as arrows on a two-dimensional plane drawn connecting the origin $(0, 0)$ and points $(x_0, x_1)$ and $(y_0, y_1)$, respectively. Matrix $A$ transforms vector $X$ into $Y$. This operation can change both the length (absolute value) of the vector and its direction.

$$
\text{Length}_X = |X| = \sqrt{x_0^2 + x_1^2} \quad \theta_X = \arcsin \left( \frac{x_1}{|X|} \right)
$$

If the direction of the vector does not change, it means that all vector components are scaled by an identical constant $\lambda$. Equation

$$
A \times X = \lambda \times X
$$

is called an eigenvalue equation (see below)
Vector Transformations

Here are other examples of important vector transformations:

Identity: keeps the vector unchanged, \( \mathbf{I} \times \mathbf{X} = \mathbf{X} \)

Scaling: scales the vector components by a constant factor, \( \lambda \mathbf{I} \times \mathbf{X} = \lambda \mathbf{X} \)

Permutation: exchanges the meaning of axes in an n-dimensional space. For example, if \( \mathbf{X} = (1,0,0) \), \( \mathbf{Y} = (0,1,0) \), and \( \mathbf{Z} = (0,0,1) \) are unit vectors in 3D space, matrix \( \mathbf{A} \) that exchanges \( \mathbf{Y} \) and \( \mathbf{Z} \) but leaves \( \mathbf{X} \) unchanged is an YZ-permutation matrix \( \mathbf{A} \times \mathbf{X} = \mathbf{X} ; \mathbf{A} \times \mathbf{Y} = \mathbf{Z} ; \mathbf{A} \times \mathbf{Z} = \mathbf{Y} \)

Rotation: rotation of the vector w/o changing its length (e.g., see this link)

This matrix is a rotation matrix for counterclockwise rotation of a vector by angle \( \theta_R \) in a two dimensional space

\[
R = \begin{pmatrix}
\cos(\theta_R) & -\sin(\theta_R) \\
\sin(\theta_R) & \cos(\theta_R)
\end{pmatrix}
\]

The length remains unchanged upon rotation

\[ |\mathbf{X}| = |\mathbf{Y}| \]
Examples of Vector Transformations

Unit vectors in 3D-space

\[
X := \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad Y := \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad Z := \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}
\]

Identity operation

\[
I := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad I \cdot X = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}
\]

Scaling by a constant \( \lambda := 5 \)

\[
\lambda \cdot I = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix} \quad \lambda \cdot I \cdot X = \begin{pmatrix} 5 \\ 0 \\ 0 \end{pmatrix}
\]

XY permutation

\[
P := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad P \cdot X = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad P \cdot Y = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad P \cdot Z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}
\]

Vector in 2D-space pointing diagonally

Rotation matrix for a 90-degree counterclockwise rotation

\[
Xd := \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad R := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad R \cdot Xd = \begin{pmatrix} -1 \\ 1 \end{pmatrix}
\]

\[ Xd = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \]

\[ R \cdot Xd = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \]
It is often necessary to solve system of \( n \) linear equations with \( n \) variables. Mathcad's ability to manipulate matrices make this very simple. For example, a system of equations:

\[
\begin{align*}
6x_1 + 2x_2 + 8x_3 &= 14 \\
x_1 + 3x_2 + 4x_3 &= 5 \\
5x_1 + 6x_2 + 2x_3 &= 7
\end{align*}
\]

can be rewritten in matrix form as follows \( M \cdot X = A \) where \( M \) is the coefficient matrix, \( X \) is the vector containing unknown variables, and \( A \) is the vector containing the right sides of the equations above. If \( M \) is \textbf{not singular}, that is \(|M|\neq0\), one can find a \textbf{unique solution} \( X \) as follows: \( X = M^{-1} \cdot A \) where \( M^{-1} \) is the inverse of matrix \( M \).

\[
M := \begin{pmatrix}
6 & 2 & 8 \\
1 & 3 & 4 \\
5 & 6 & 2
\end{pmatrix}, \quad A := \begin{pmatrix}
14 \\
5 \\
7
\end{pmatrix}
\]

\[
\text{Define the coefficient matrix and the result vector.}
\]
\[
\text{Note that the solution will exist only if } M \text{ is not a singular matrix (determinant must be not equal to 0)}
\]

\[
\text{Verify that } M \text{ is not singular:} \quad |M| = -144
\]

\[
\text{Find inverse of matrix } M: \quad \text{Inv}_M := M^{-1}
\]

\[
\text{Inv}_M = \begin{pmatrix}
0.125 & -0.306 & 0.111 \\
-0.125 & 0.194 & 0.111 \\
0.063 & 0.181 & -0.111
\end{pmatrix}
\]

\[
\text{Find the solution for } X: \quad X := \text{Inv}_M \cdot A
\]

\[
X = \begin{pmatrix}
1 \\
0 \\
1
\end{pmatrix}
\]

\textbf{Answer: } x_1=1; \ x_2=0; \ x_3=1
Imagine you have the following system of two equations with two unknown variables $x_1$ and $x_2$, and 6 known constants $a_{11}$, $a_{12}$, $a_{21}$, $a_{22}$, $b_1$, $b_2$:

$$a_{11}x_1 + a_{12}x_2 = b_1$$
$$a_{21}x_1 + a_{22}x_2 = b_2$$

This system is easy to solve by explicit substitution. The solution is:

$$x_1 = \frac{a_{22}b_1 - a_{12}b_2}{a_{22}a_{11} - a_{12}a_{21}}$$
$$x_2 = \frac{a_{11}b_2 - a_{21}b_1}{a_{22}a_{11} - a_{12}a_{21}}$$

Note that this solution exists as along as the denominator in the above equations is not zero. This denominator happens to be nothing else than the determinant of matrix $A$:

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad |A| = a_{11}a_{22} - a_{12}a_{21}$$

PRACTICE: solve the following system of linear equations in Mathcad using both explicit formulae given above and the Mathcad's matrix manipulation tools. Compare your answers.

$$5x_1 + 4x_2 - 1 = 0$$
$$2 - 3x_1 + 4x_2 = 0$$
How Does It Work?

Because resistors $R_1$ and $R_{\text{POT}}$ come together at point $c$, and we know that the voltages at $a$ and $b$ are the same, there must be the same voltage drop across $R_1$ and $R_{\text{POT}}$ (not the same current). Thus, we have

$$i_3 \cdot R_1 = i_2 \cdot R_{\text{POT}}.$$

Similarly, because $R_2$ and $R_{\text{RTD}}$ are connected at point $d$, we can say that

$$i_3 \cdot R_2 = i_2 \cdot R_{\text{RTD}}.$$

If you solve for $i_3$ in one equation and substitute into the other, you get

$$R_{\text{RTD}} = R_{\text{POT}} \frac{R_2}{R_1}.$$

In this way, if you know $R_1$ and $R_2$ and have a reading on the potentiometer, you can calculate $R_{\text{RTD}}$.

(a) Given the following resistances, what is $R_{\text{RTD}}$? Assume that a 9-volt battery is used for $E$.

$$R_0 = 20 \text{ ohms};$$
$$R_1 = 10 \text{ ohms};$$
$$R_2 = 5 \text{ ohms};$$
$$R_{\text{POT}} \text{ adjusted to } 12.3 \text{ ohms}.$$

(b) Use Kirchhoff’s current law at either point $c$ or $d$, and Kirchhoff’s voltage law to determine the values of $i_1$, $i_2$, and $i_3$. 

Eigenvalues an Eigenvectors

In physical chemistry applications, one frequently needs to solve the eigenvalue equation

$$MV = \lambda V \quad \text{or} \quad (M - \lambda I)V = 0$$

where $M$ is a square matrix, and $I$ is the identity matrix of the same size. A vector $V$ satisfying this equation is called an eigenvector, and the corresponding constant $\lambda$ is called an eigenvalue. Application of matrix $M$ does not change the direction of $V$; it only scales it up or down by a constant factor $\lambda$. In physical chemistry, one is often interested in finding ALL possible eigenvalues and eigenvectors for a given matrix. The trivial solution $V=0$ is of no interest, as it always satisfies the eigenvalue equation. Non-trivial solution exists only if the determinant of matrix $(M-\lambda V)$ is equal to zero. The equation

$$|M - \lambda I| = 0$$

is called characteristic equation. For a $n \times n$ matrix, this equation is a polynomial in $\lambda$ of degree $n$. Therefore, there are exactly $n$ possible eigenvalues (and corresponding eigenvectors).

Mathcad provides the following capabilities for solving the eigenvalue problem:

- **eigenvals**(M) – returns a vector containing eigenvalues of square matrix M
- **eigenvect**(M,z) – returns a single eigenvector corresponding to a given eigenvalue z
- **eigenvecs**(M) - returns all eigenvectors arranged in a matrix, wherein $i^{\text{th}}$ column of the matrix corresponds to the $i^{\text{th}}$ element of the vector returned by eigenvals(M)
Eigenvalues an Eigenvectors

**Example:** Find eigenvalues and eigenvectors for matrix $P$ below. Verify that your solutions satisfy the eigenvalue equation by explicit substitution.

1. Define a square matrix

$$ P := \begin{pmatrix} 0 & 2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} $$

2. Calculate all possible eigenvalues

$$ \text{eigenvals}(P) = \begin{pmatrix} 2 \\ -2 \\ 1 \end{pmatrix} \quad \text{eigenvecs}(P) = \begin{pmatrix} 0.707 & -0.707 & 0 \\ 0.707 & 0.707 & 0 \\ 0 & 0 & 1 \end{pmatrix} $$

3. Calculate the corresponding eigenvectors

$$ \text{value1} := \text{eigenvals}(P)_0 = 2 \quad \text{Here we pulled the first eigenvalue out of the vector returned by function eigenvals}(P) $$

$$ \text{vector1} := \text{eigenvecs}(P)\langle 0 \rangle = \begin{pmatrix} 0.707 \\ 0.707 \\ 0 \end{pmatrix} \quad \text{... and the first eigenvector returned by eigenvecs}(P) $$

$$ P\cdot\text{vector1} - \text{value1}\cdot\text{vector1} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{... and then explicitly verified that they satisfy the eigenvalue equation} $$
Sorting Functions

- \texttt{length(v)} – returns the number of elements in vector \( v \)
- \texttt{last(v)} – returns the index of the last element in vector \( v \) (depends on the value of ORIGIN)
- \texttt{cols(A)} and \texttt{rows(A)} return the number of columns and rows in matrix \( A \)
- \texttt{max(A)} and \texttt{max(A)} return the maximum and minimum values in matrix or vector \( A \)
- \texttt{sort(v)} – sorts elements of vector \( v \) into ascending order
- \texttt{reverse(v)} – reverses the order of elements in vector \( v \)
- \texttt{csort(M, n)} – sorts elements in column \( n \) or matrix \( A \) in an ascending order
- \texttt{rsort(M, n)} – sorts elements in row \( n \) or matrix \( A \) in an ascending order

\textbf{Example:} For the following grade distribution in Chem 5, calculate the maximum, minimum, median, and average grades.

Grades are entered manually as a row-vector in order to save space:

\[ \text{Gr} := (2.2 \ 1.9 \ 1.6 \ 1.2 \ 3.8 \ 2.6 \ 3.9 \ 2.1 \ 1.5 \ 2.7 \ 3.8 \ 4.0 \ 3.8 \ 2.2 \ 3.5 \ 3.4 \ 1.1) \]

\[ \text{Gr} := \text{Gr}^T \]

Gr is transposed so that it becomes a proper column vector. Note that operations like \texttt{length(v)} only work on column vectors!

- \texttt{length(Gr)} = 17
- \texttt{min(Gr)} = 1.1
- \texttt{max(Gr)} = 4
- \texttt{mean(Gr)} = 2.665
- \texttt{median(Gr)} = 2.6

The number of elements in this vector is calculated

\[ \text{The worst and best grades in this class} \]

\[ \text{The average grade is calculated using function \texttt{mean}(x,y,z,...)} \]

\[ \text{Function \texttt{median}(x,y,z,...) returns the median value of the grade} \]

Alternatively we can calculate the median as the middle element of the sorted vector

\[ \text{Sorted := sort(Gr)} \]

\[ \text{index := floor}\left(\frac{\text{length(Gr)}}{2}\right) = 8 \]

\[ \text{Sorted}_{\text{index}} = 2.6 \]
Resizing Matrices

- **submatrix**\((A, r_{\text{start}}, r_{\text{stop}}, c_{\text{start}}, c_{\text{stop}})\) – returns a part of matrix \(A\) contained between the specified rows and columns
- **augment**() – combines two arrays side by side
- **stack**() places one array on top of another
- **< >** column operator selects a specific column of matrix \(A\)

\[
\begin{align*}
&\text{fl}(rr, cc) := rr + cc^2 \\
&M1 := \text{matrix}(3, 3, f1) = \begin{pmatrix} 0 & 1 & 4 \\ 1 & 2 & 5 \\ 2 & 3 & 6 \end{pmatrix} \\
&M2 := \text{matrix}(3, 3, f2) = \begin{pmatrix} 0 & -1 & -4 \\ 1 & 0 & -3 \\ 2 & 1 & -2 \end{pmatrix} \\
&\text{stack}(M1, M2) = \begin{pmatrix} 0 & 1 & 4 \\ 1 & 2 & 5 \\ 2 & 3 & 6 \\ 0 & -1 & -4 \\ 1 & 0 & -3 \\ 2 & 1 & -2 \end{pmatrix} \\
&\text{augment}(M1, M2) = \begin{pmatrix} 0 & 1 & 4 & 0 & -1 & -4 \\ 1 & 2 & 5 & 1 & 0 & -3 \\ 2 & 3 & 6 & 2 & 1 & -2 \end{pmatrix} \\
&\text{submatrix}(M1, 0, 1, 0, 1) = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix} \\
&M1^{\langle 0 \rangle} = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \\
&M1^{\langle 1 \rangle} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \\
&M1^{\langle 2 \rangle} = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}
\end{align*}
\]