|  |  |  |
| --- | --- | --- |
|  | FACULTY OF ENGINEERING TECHNOLOGY, UNIVERSITI TEKNIKAL MALAYSIA MELAKA | |
| DISCRETE MATHEMATICS | | |
| BEEC 3413 | SEMESTER 1 | SESI 2021/2022 |
| LAB 2: MATRIX | | |
| NAME OF GROUP MEMBERS AND MATRIX NUMBER | 1.RHAMAN KAZI ASHIKUR | B081910450 |
| 2. |  |
| 3. |  |
| COURSE | BEEC | |
| DATE | 3/11/ 2021 | |
| NAME OF INSTRUCTOR | AZMAN AWANG TEH | |
| EXAMINER’S COMMENT | | VERIFICATION STAMP |
| TOTAL MARKS |

## OBJECTIVES

1.To understand programming of **matrix** in form of arrays. 2.To make program for matrix operations.

## EQUIPMENTS

1. Personal Computer.
2. R Software.
3. SYNOPSIS & THEORY
   1. MATRIX

**Definition 1.** A matrix is a rectangular array of numbers. A matrix with *m* rows and *n* columns is called an *m × n* matrix. The plural of matrix is matrices. A matrix with the same number of rows as columns is called square. Two matrices are equal if they have the same number of rows and the same number of columns and the corresponding entries in every position are equal.

**Definition 2.** Let *m* and *n* be positive integers and let

The *i* th row of *A* is the 1 *× n* matrix. The *j* th column of *A* is the *m ×* 1 matrix

The (*i* , *j* )th element or entry of *A* is the element *ai j* , that is, the number in the *i* th row and *j* th column of *A*. A convenient shorthand notation for expressing the matrix *A* is to write *A =* [*ai j* ], which indicates that *A* is the matrix with its (*i* , *j* )th element equal to *ai j* .

* 1. MATRIX ARITHMETIC

The basic operations of matrix arithmetic will now be discussed, beginning with a definition of matrix addition.

**Definition 3.** Let *A =* [*ai j* ] and *B =* [*bi j* ] be *m × n* matrices. The sum of *A* and *B* , denoted by *A + B* , is the *m × n* matrix that has *ai j + bi j* as its (*i* , *j* )th element. In other words, *A + B =* [*ai j + bi j* ].

**Definition 4.** Let *A* be an *m × k* matrix and *B* be a *k × n* matrix. The product of *A* and *B* , denoted by *AB* , is the *m × n* matrix with its (*i* , *j* )th entry equal to the sum of the products of the corresponding elements from the *i* th row of *A* and the *j* th column of *B* . In other words, if *AB =* [*ci j* ], then

*ci j = ai* 1*b*1*j + ai* 2*b*2*j + · · · + ai k bk j* . (3.2)

In general, suppose that *A* is an *m ×n* matrix and *B* is an *r × s* matrix. Then *AB* is defined only when *n = r* and *B A* is defined only when *s = m*.

* 1. TRANSPOSE OF MATRICES

**Definition 5.** Let *A =* [*ai j* ] be an *m ×n* matrix. The transpose of *A*, denoted by *At* , is the *n ×m* matrix obtained by interchanging the rows and columns of *A*. In other words, if *At =* [*bi j* ], then *bi j = a j i* for *i =* 1, 2, *· · ·* , *n* and *j =* 1, 2, *· · ·* , *m*.

* 1. ARRAY AND MATRICES IN R

An array can be considered as a multiply subscripted collection of data entries, for example numeric. R allows simple facilities for creating and handling arrays, and in particular the spe- cial case of matrices.

A dimension vector is a vector of non-negative integers. If its length is *k* then the array is *k*-dimensional, e.g. a matrix is a 2-dimensional array. The dimensions are indexed from one up to the values given in the dimension vector.

A vector can be used by R as an array only if it has a dimension vector as its dim attribute. Suppose, for example, *z* is a vector of 1500 elements. The assignment

# > dim(z) <- c(3,5,100)

gives it the dim attribute that allows it to be treated as a 3 by 5 by 100 array.

* 1. EXAMPLE OF ARRAY AND MATRICES IN R

# > x <- array(1:20, dim=c(4,5))

Generate a 4 by 5 array with elements are 1 to 20. and the result is

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| > x | [,1] | [,2] | [,3] | [,4] | [,5] |
| [1,] | 1 | 5 | 9 | 13 | 17 |
| [2,] | 2 | 6 | 10 | 14 | 18 |
| [3,] | 3 | 7 | 11 | 15 | 19 |
| [4,] | 4 | 8 | 12 | 16 | 20 |

# > A = matrix(

+ c(2, 4, 3, 1, 5, 7),nrow=2,ncol=3)

Generate a 2 by 3 matrix with elements are 2, 4, 3, 1, 5, 7. and the result is

# > A

[,1] [,2] [,3]

[1,] 2 3 5

[2,] 4 1 7

By default the element will fill by column first, if we need to change into by row we should add byrow=TRUE.

# > A = matrix(

+ c(2, 4, 3, 1, 5, 7),nrow=2,ncol=3,byrow=TRUE)

and the result is

# A

[,1] [,2] [,3]

[1,] 2 4 3

[2,] 1 5 7

* 1. LOOPING

R has three statements that provide explicit looping. They are for, while and repeat. The two built-in constructs, next and break, provide additional control over the evaluation. Each of the three statements returns the value of the last statement that was evaluated. It is possi- ble, although uncommon, to assign the result of one of these statements to a symbol.

The value returned by a loop statement is always NULL and is returned invisibly.

3.6.1 FOR

The syntax of the for loop is

# for ( name in vector ) statement1

where *vector* can be either a vector or a list. For each element in *vector* the variable *name* is set to the value of that element and *statement1* is evaluated. A side effect is that the variable *name* still exists after the loop has concluded and it has the value of the last element of *vector* that the loop was evaluated for.

example:

# > for (i in 1:50 ) {

+ a=a+i^2

+ }

## PROCEDURE

4.1 MATRIX OPERATION

1. Open R Gui, and create a new script.

2. Explore the syntaxes of matrix and array in R.

3. Define new matrices.

using R.

1. If *X* , and *Y* are matrices, then create functions for *X + Y* , *X Y* , and *X t* using looping and then calculate
   * *A + A*
   * *A + B*
   * *CC*
   * *C A*
   * *At A*

## RESULT

**Coding part**

|  |
| --- |
| A <- matrix(c (1,4,5,3,2,6,2,0,8,-4,7,-3),nrow=3,ncol=4)  B <- matrix(c (-1,12,3,8,6,3,7,4),nrow=4,ncol=2)  C <- matrix(c (5,-13,9,7,11,8,9,2,-7),nrow=3,ncol=3)  mplus <- function(x,y){    r1=nrow(x)  r2=nrow(y)  c1=ncol(x)  c2=ncol(y)    if (r1!=r2 || c1!=c2){  print('Sorry, your matrix size does not match')  return()  }  z = matrix(0,nrow=r1,ncol=c1)  for (i in 1:r1){  for (j in 1:c1){  z[i,j]=x[i,j]+y[i,j]  }  }  return(z)  }  mMultiply <- function(x,y){    r1=nrow(x)  r2=nrow(y)  c1=ncol(x)  c2=ncol(y)  if (r2 != c1){  print('size does not match')  return()  }  z = matrix(0,nrow = r1, ncol = c2)    if(r1 == r2 && c1 == c2) { # This is for matrices that have identical dimension  for (i in 1:r1){  for (j in 1:c1){  z[i,j]=x[i,j] \* y[i,j]  }  }  } else if(c1 == r2) { # This is for C \* A and A' \* A  for (i in 1:r1){  for (j in 1:c2){  for (k in 1:r2) {  z[i,j] = z[i,j] + (x[i,k] \* y[k,j])  }  }  }  }    return(z)  }  transpose <- function(x) { # function for transpose  r1 = nrow(x)  c1 = ncol(x)    z = matrix(0, nrow = c1, ncol = r1)    for (i in 1:nrow(x))  {  for (j in 1:ncol(x))  {  z[j, i] = x[i, j]  }  }  return(z)  }  cat("This is for A + A\n")  print(mplus(A,A))  cat("\nThis is for A + B\n")  print(mplus(A,B))  cat("\nThis is for C \* C\n")  print(mMultiply(C,C))  cat("\nThis is for C \* A\n")  print(mMultiply(C,A))  cat("\nThis is for A' \* A\n")  print(mMultiply(transpose(A), A)) |

* + ***A + A***

Text

Description automatically generated with medium confidence

* + *A + B*

Graphical user interface, text, application, email

Description automatically generated

* + ***CC***

Text

Description automatically generated with medium confidence

* + ***C A***

Text

Description automatically generated

* + ***At A***

A picture containing text

Description automatically generated

## DISCUSSION

A matrix is a collection of numbers arranged in a row-by-row and column-by-column arrangement.

The elements of a matrix must be enclosed in parenthesis or brackets.

Example –

The 3 \* 3 matrix means a matrix with 3 rows & 3 columns with a total of 9 elements. (3\*3 = 9)

**Matrix Sums**

The sum A + B of two matrices A, B (which must have the same number of rows, and the same number of columns) is the matrix (also with the same shape) given by adding corresponding elements of A and B. A + B = [aij + bij]

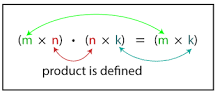
**Matrix Sums Algorithm**

|  |
| --- |
| mplus <- function(x,y){    r1=nrow(x)  r2=nrow(y)  c1=ncol(x)  c2=ncol(y)    if (r1!=r2 || c1!=c2){  print('Sorry, your matrix size does not match')  return()  }  z = matrix(0,nrow=r1,ncol=c1)  for (i in 1:r1){  for (j in 1:c1){  z[i,j]=x[i,j]+y[i,j]  }  }  return(z)  } |

**Matrix multiplication**

What is matrix multiplication in discrete mathematics?

In mathematics, matrix multiplication is different from the multiplication that we perform, generally. It is a binary operation that performs between two matrices and produces a new matrix

[](https://www.google.com/search?q=What+is+matrix+multiplication+in+discrete+mathematics?&sa=X&bih=641&biw=1422&rlz=1C1ONGR_enMY932MY932&hl=en&sxsrf=AOaemvJVa69G6CVvums7ntkR_gWvYp7z_g:1635941400455&tbm=isch&source=iu&ictx=1&fir=Sl6Ik8aRwp7HwM%252CpWS-Fw_XYIR1IM%252C_&vet=1&usg=AI4_-kRQtEnPBQiS5PR7uaDi4QeYz8N0Tg&ved=2ahUKEwjFlLDQlPzzAhUyIbcAHelmByEQ9QF6BAgUEAE#imgrc=Sl6Ik8aRwp7HwM)

**Matrix multiplication Algorithm**

|  |
| --- |
| mMultiply <- function(x,y){    r1=nrow(x)  r2=nrow(y)  c1=ncol(x)  c2=ncol(y)  if (r2 != c1){  print('size does not match')  return()  }  z = matrix(0,nrow = r1, ncol = c2)    if(r1 == r2 && c1 == c2) { # This is for matrices that have identical dimension  for (i in 1:r1){  for (j in 1:c1){  z[i,j]=x[i,j] \* y[i,j]  }  }  } else if(c1 == r2) { # This is for C \* A and A' \* A  for (i in 1:r1){  for (j in 1:c2){  for (k in 1:r2) {  z[i,j] = z[i,j] + (x[i,k] \* y[k,j])  }  }  }  }    return(z) |

**Matrix Transposition**

If A = [aij] is an m × n matrix, the transpose of A

(often written A

or AT) is the n × m matrix given by

At = B = [bij] = [aij] (1 ≤ i ≤ n,1 ≤ j ≤ m)

A picture containing text, clock

Description automatically generated

**Matrix Transposition Algorithm**

|  |
| --- |
| transpose <- function(x) { # function for transpose  r1 = nrow(x)  c1 = ncol(x)    z = matrix(0, nrow = c1, ncol = r1)    for (i in 1:nrow(x))  {  for (j in 1:ncol(x))  {  z[j, i] = x[i, j]  }  }  return(z)  } |

## CONCLUSION

From this LAB I understand programming of **matrix** in form of arrays and make program for matrix operations using R programming

## REFERENCE

The R Manuals, <http://www.r-project.org/>