University of New Mexico

An Integrated Maple Package for Algebraic Interval Neutrosophic Matrices

Said Broumi[1,2,*], Mohamed Bisher Zeina[3], Mohammad Abobala[4]

[1] Laboratory of Information Processing, Faculty of Science Ben M'Sik, University of Hassan II, Casablanca, Morocco

broumisaid78@gmail.com

[2]Regional Center for the Professions of Education and Training (C.R.M.E.F), Casablanca-Settat, Morocco.

[3] Dept. of Mathematical Statistics, Faculty of Science, University of Aleppo, Aleppo, Syria

bisher.zeina@gmail.com

[4] Tishreen University, Faculty off Science, Department of Mathematics, Latakia, Syria

mohammadabobala777@gmail.com

Abstract: In this paper, a maple code is presented to do many operations on interval valued neutrosophic matrices including entering the elements of the matrix, checking whether a given matrix is an interval valued neutrosophic matrix or not, finding complement of an interval valued neutrosophic matrix, finding score, accuracy, and certainty measures, union and intersection of two interval valued neutrosophic matrices, sum and product of two interval valued neutrosophic matrices and finding the transpose of a given interval valued of neutrosophic matrix. What distinguishes this code is its simplicity to understand and to call the functions. Many examples are presented and solved successfully.

Keywords: Maple; Neutrosophic Set; Single Valued Neutrosophic Set; Interval Valued Neutrosophic Set; Operations on Matrices.

1. Introduction

Fuzzy Sets were presented by Zadeh [1] to expand the concept of crisp sets allowing elements to belong to the sets partially (with membership degree between 0 and 1), then the last concept expanded by Atanassov [2] to what is known by intuitionistic fuzzy sets adding nonmebmbership component to describe elements of sets. In 1995, Smarandache [3] presented neutrosophic sets as an extension of fuzzy sets and intuitionistic fuzzy sets in which each element is described by three independent componentsl; truth, indeterminacy and false memberships. Many other extensions to neutrosophic sets were presented including refined neutrosophic sets, interval valued neutrosophic sets, bipolar neutrosophic sets, generalized neutrosophic sets, neutrosophic vague soft expert set, fermatean neutrosophic sets, etc.

Many mathematical studies were done on neutrosophic sets and many branches of mathematics were extended to the new concept of logic including probabiliy theory, operations research, statistics, linear algebra, abstract algebra, queueuing theory, artificial intelligence and data minig.[4-14]

Since delaing with neutrosophic sets is very complex and operations on it take long time , then many researchers wrote programming packages and codes to make dealing with it more simple.

Salama et al.[15] presented and introduction to develop programming softwares to deal with neutrosophic sets. Bakro et al.[16] wrote a matlab code to neutrosophication functions and their implementation. Broumi et al.[17] wrote a matlab code to implement neutrosophic membership functions and graphing it. Bisher Zeina et al.[18] presented a maple package to do operations on single valued neutrosophic sets using α,β,γ-Cuts, Broumi et al.[19] wrote a maple package to perform operations on single valued neutrosophic matrices. In this paper we generalize the code presented in [19] to deal with interval valued neutrosphic matrices and do operations on it.

2. Background on Neutrosophic Sets

Definition 2.1[20]

Let Ω be a universe, we call $A \subseteq \Omega$ a neutrosophic set if elements of A are described by their membership degree $T_A(x)$, nonmembership degree $F_A(x)$ and indeterminacy degree $I_A(x)$ and we denote that by:

$$A = \{< T_A(x), I_A(x), F_A(x) > \; ; x \in \Omega\}$$

Where:

$$T_A(x), \mathrm{I_A}(x), F_A(x) \in]^{-}0, 1^{+}[\;\; \& \;\; ^{-}0 \;\; \leq T_A(x) + I_A(x) + F_A(x) \leq \;\; 3^{+}$$

Definition 2.2 [20]

Let Ω be a universe, we call $A \subseteq \Omega$ a single valued neutrosophic set if:

$$A = \{< T_A(x), I_A(x), F_A(x) > \; ; x \in \Omega\}$$

Where:

$$T_A(x), \mathrm{I_A}(x), F_A(x) \in [0,1] \;\; \& \;\; 0 \;\; \leq T_A(x) + I_A(x) + F_A(x) \leq \;\; 3$$

Definition 2.3 [21]

Let Ω be a universe, we call $A \subseteq \Omega$ an interval valued neutrosophic set if:

$$A = \{< [T_A^L(x), T_A^U(x)], [I_A^L(x), I_A^U(x)], [F_A^L(x), F_A^U(x)] > \; ; \;\; \mathrm{x} \in \Omega\}$$

Where:

$$T_A^L(x), T_A^U(x), I_A^L(x), I_A^U(x), F_A^L(x), F_A^U(x) \in [0,1]$$

Definition 2.4

Interval valued neutrosophic matrix of order $m \times n$ is defined as follows:

$$A = \left[< a_{ij}, \left[Ta_{ij}^L, Ta_{ij}^U\right], \left[Ia_{ij}^L, Ia_{ij}^U\right], \left[Fa_{ij}^L, Fa_{ij}^U\right] >\right]_{m\times n}$$

3. Maple Package to Do Operations on Interval Valued Neutrosophic Matrices

3.1. Entering Interval Valued Neutrosophic Matrices

To enter interval valued neutrosophic matrix we call the function IVNIInput(m,n) where m, n are numbers of rows and columns respectively and the written function is as follows:

```
restart;interface(warnlevel=0);
```

```
with(Maplets[Elements]):

with(Maplets):
IVNIInput:=proc(m::integer,n::integer)

local mat:=Matrix(m,n);

for i from 1 to m by 1 do

for j from 1 to n by 1 do

truthL:=Maplet(InputDialog['x'](cat("Enter lower truth of element
",i,",",j),'onapprove'=Shutdown(['x']),'oncancel'=Shutdown()));

truthL:=Display(truthL);

truthL:=parse(op(truthL));

truthU:=Maplet(InputDialog['x'](cat("Enter upper truth of element
",i,",",j),'onapprove'=Shutdown(['x']),'oncancel'=Shutdown()));

truthU:=Display(truthU);

truthU:=parse(op(truthU));

indeterminacyL:=Maplet(InputDialog['x'](cat("Enter lower indeterminacy of element
",i,",",j),'onapprove'=Shutdown(['x']),'oncancel'=Shutdown()));

indeterminacyL:=Display(indeterminacyL);

indeterminacyL:=parse(op(indeterminacyL));

indeterminacyU:=Maplet(InputDialog['x'](cat("Enter upper indeterminacy of element
",i,",",j),'onapprove'=Shutdown(['x']),'oncancel'=Shutdown()));

indeterminacyU:=Display(indeterminacyU);

indeterminacyU:=parse(op(indeterminacyU));

falsityL:=Maplet(InputDialog['x'](cat("Enter lower falsity of element
",i,",",j),'onapprove'=Shutdown(['x']),'oncancel'=Shutdown()));

falsityL:=Display(falsityL);

falsityL:=parse(op(falsityL));

falsityU:=Maplet(InputDialog['x'](cat("Enter upper falsity of element
",i,",",j),'onapprove'=Shutdown(['x']),'oncancel'=Shutdown()));

falsityU:=Display(falsityU);

falsityU:=parse(op(falsityU));

mat(i,j):=convert([[truthL,truthU],[indeterminacyL,indeterminacyU],[falsityL,falsityU]],string);

end do;
```

```
end do;
mat;
end proc:
```

3.2. Checking whether the matrix is IVNM or not

We can call the function **IVNCheck (mat)** defined below to check whether matrix mat is interval valued neutrosophic matrix or not:

```
IVNCheck:=proc(mat)
IsMembership:=proc(num)
if num<0 or num>1 then return false else return true end if;
end proc:
m,n:=LinearAlgebra[Dimension](mat);
result:=true;
for i from 1 to m by 1 do
for j from 1 to n by 1 do
x:=parse(mat(i,j));
truth:=x[1];
indeterminacy:=x[2];
falsity:=x[3];
truthL:=truth[1];
truthU:=truth[2];
indeterminacyL:=indeterminacy[1];
indeterminacyU:=indeterminacy[2];
falsityL:=falsity[1];
falsityU:=falsity[2];
result:= IsMembership(truthL) and IsMembership(truthU) and IsMembership(indeterminacyL) and
IsMembership(indeterminacyU) and IsMembership(falsityL) and IsMembership(falsityU);
if not result then break; end if;
end do;
if not result then break; end if;
```

```
end do;
if result then cat("your matrix is an interval valued neutrosophic matrix") else cat("your matrix is not a single valued neutrosophic matrix") end if;
end proc:
```

Example 1. In this example we define an interval valued neutrosophic matrix E and check whether it is right defined or not where:

E=

$$\begin{pmatrix} <[1,1],[.7,.8],[.1,.6]> & <[.2,.4],[.2,.8],[.1,.9]> \\ <[.8,.9],[.3,.5],[.1,.2]> & <[.1,.2],[.5,.7],[.2,.5]> \end{pmatrix}$$

The interval valued neutrosophic matrix E can be inputted in Maple like this:

E := Matrix(2, 2, [["[[1, 1], [.7, .8], [.1, .6]]", "[[.2, .4], [.2, .8], [.1, .9]]"], ["[[.8, .9], [.3, .5], [.1, .2]]", "[[.1, .2], [.5, .7], [.2, .5]]"]]);

Or like this:

x:= IVNIInput (2,2);

Then an input box dialogue is going to appear and lead you how to input elements.

Result of checking whether matrix E is Interval-Valued Neutrosophic Matrix or not can be obtained by calling the command IVNCheck(E);

And the result will be:

"your matrix is an interval valued neutrosophic matrix"

3.3. Finding complement of interval valued neutrosophic matrix

For a given IVNM $A = \left[< a_{ij}, \left[Ta_{ij}^L, Ta_{ij}^U\right], \left[Ia_{ij}^L, Ia_{ij}^U\right], \left[Fa_{ij}^L, Fa_{ij}^U\right] >\right]_{m\times n}$, the complement of A is defined as follow:

$$A^c = A = \left[< a_{ij}, \left[Fa_{ij}^L, Fa_{ij}^U\right], \left[1 - Ia_{ij}^U, 1 - Ia_{ij}^L\right], \left[Ta_{ij}^L, Ta_{ij}^U\right] >\right]_{m\times n} \quad (10)$$

To find the complement of interval valued neutrosophic matrix we can call the function **IVNMCompelementOf(mat)** which is defined as follow:

```
IVNMCompelementOf:=proc(mat::Matrix)
temp:=LinearAlgebra[Copy](mat);
m,n:=LinearAlgebra[Dimension](temp);
```

```
for i from 1 to m by 1 do
for j from 1 to n by 1 do
x:=parse(mat(i,j));
falsity:=x[1];
indeterminacy:=x[2];
truth:=x[3];
indeterminacyL:=1-indeterminacy[2];
indeterminacyU:=1-indeterminacy[1];
temp(i,j):=convert([truth,[indeterminacyL,indeterminacyU],falsity],string);
end do;
end do;
temp;
end proc:
```

Example 2. find the complement of matrix E in example 1.

the complement of matrix E is:

$$E^c = \begin{pmatrix} < [.1,.6],[.2,.3],[1,1] > & < [.1,.9],[.2,.8],[.2,.4] > \\ < [.1,.2],[.5,.7],[.8,.9] > & < [.2,.5],[.3,.5],[.1,.2] > \end{pmatrix}$$

By calling the function

SVNMCompelementOf1(E);

Same results appear:

$$\begin{bmatrix} \text{"[[.1, .6], [.2, .3], [1, 1]]"} & \text{"[[.1, .9], [.2, .8], [.2, .4]]"} \\ \text{"[[.1, .2], [.5, .7], [.8, .9]]"} & \text{"[[.2, .5], [.3, .5], [.1, .2]]"} \end{bmatrix}$$

3.4. Finding score, accuracy and certainty matrices of interval valued neutrosophic matrices

Suppose A is an interval neutrosophic matrix, then score, accuracy and certainty measures are defined as follows: [22]

$$\tilde{S}_{IVNN}(x)=\frac{T_A^L(x)+T_A^U(x)+4-I_A^L(x)-I_A^U(x)-F_A^L(x)-F_A^U(x)}{6}$$

$$\tilde{S}_{SVNN}(x)=\frac{2+T_A(x)-I_A(x)-F_A(x)}{3}$$

$$\tilde{A}_{IVNN}(x)=\frac{T_A^L(x)+T_A^U(x)-F_A^L(x)-F_A^U(x)}{2}$$

$\tilde{A}_{SVNN}(x)$=$T_A(x) - F_A(x)$

$\tilde{C}_{IVNN}(x)$=$\frac{T_A^L(x)+T_A^U(x)}{2}$

$\tilde{C}_{SVNN}(x)$=$T_A(x)$

Three maple functions **ScoreMatrix( )**, **AccuracyMatrix ( )** and **CertaintyMatrix ( )** are defined as follows:

```
ScoreMatrix:=proc(mat::Matrix)
m,n:=LinearAlgebra[Dimension](mat);
scoreMat:=Matrix(m,n);
for i from 1 to m by 1 do
for j from 1 to n by 1 do
x:=parse(mat(i,j));
truth:=x[1];
indeterminacy:=x[2];
falsity:=x[3];
truthL:=truth[1];
truthU:=truth[2];
indeterminacyL:=indeterminacy[1];
indeterminacyU:=indeterminacy[2];
falsityL:=falsity[1];
falsityU:=falsity[2];
score:=(4+truthL+truthU-indeterminacyL-indeterminacyU-falsityL-falsityU)/6;
scoreMat(i,j):=score;
end do;
end do;
scoreMat;
end proc:
AccuracyMatrix:=proc(mat::Matrix)
m,n:=LinearAlgebra[Dimension](mat);
aMat:=Matrix(m,n);
```

```
for i from 1 to m by 1 do
for j from 1 to n by 1 do
x:=parse(mat(i,j));
truth:=x[1];
indeterminacy:=x[2];
falsity:=x[3];
truthL:=truth[1];
truthU:=truth[2];
indeterminacyL:=indeterminacy[1];
indeterminacyU:=indeterminacy[2];
falsityL:=falsity[1];
falsityU:=falsity[2];
a:=(truthL+truthU-falsityL-falsityU)/2;
aMat(i,j):=a;
end do;
end do;
aMat;
end proc:
CertaintyMatrix:=proc(mat::Matrix)
m,n:=LinearAlgebra[Dimension](mat);
cMat:=Matrix(m,n);
for i from 1 to m by 1 do
for j from 1 to n by 1 do
x:=parse(mat(i,j));
truth:=x[1];
indeterminacy:=x[2];
falsity:=x[3];
truthL:=truth[1];
truthU:=truth[2];
indeterminacyL:=indeterminacy[1];
```

```
indeterminacyU:=indeterminacy[2];
falsityL:=falsity[1];
falsityU:=falsity[2];
c:=(truthL+truthU)/2;
cMat(i,j):=c;
end do;
end do;
cMat;
end proc:
```

and by calling the previous three functions we get:

ScoreMatrix(E);

$$\begin{bmatrix} 0.6333333333 & 0.4333333334 \\ 0.7666666667 & 0.4000000001 \end{bmatrix}$$

AccuracyMatrix(E);

$$\begin{bmatrix} 0.6500000000 & -0.2000000000 \\ 0.7000000000 & -0.2000000000 \end{bmatrix}$$

CertaintyMatrix(E);

$$\begin{bmatrix} 1 & 0.3000000000 \\ 0.8500000000 & 0.1500000000 \end{bmatrix}$$

3.5. Computing union of two interval valued neutrosophic matrices

Union of two interval valued neutrosophic matrices A and B is defined as follow:

$A \cup B = C = [< c_{ij_T}, c_{ij_I}, c_{ij_F} >]_{m\times\ n}$

where

$c_{ij_T} = \left[a_{ij_{TL}} \vee b_{ij_{TL}}, a_{ij_{TU}} \vee b_{ij_{TU}}\ \right],$

$c_{ij_I} = \left[a_{ij_{IL}} \vee b_{ij_{IL}}, a_{ij_{IU}} \vee b_{ij_{IU}}\ \right],$

$c_{ij_F} = \left[a_{ij_{FL}} \vee b_{ij_{FL}}, a_{ij_{FU}} \vee b_{ij_{FU}}\ \right]$

And it can be evaluated by calling the function **Union(A, B)** described as follows:

```
Union:=proc(mat1::Matrix,mat2::Matrix)
m1,n1:=LinearAlgebra[Dimension](mat1);
m2,n2:=LinearAlgebra[Dimension](mat2);
if (n1=n2) and (m1=m2) then
m:=m1;n:=n1;
unionMat:=Matrix(m,n);
for i from 1 to m by 1 do
for j from 1 to n by 1 do
x:=parse(mat1(i,j));
y:=parse(mat2(i,j));
truth1:=x[1];
indeterminacy1:=x[2];
falsity1:=x[3];
truthL1:=truth1[1];
truthU1:=truth1[2];
indeterminacyL1:=indeterminacy1[1];
indeterminacyU1:=indeterminacy1[2];
falsityL1:=falsity1[1];
falsityU1:=falsity1[2];
truth2:=y[1];
indeterminacy2:=y[2];
falsity2:=y[3];
truthL2:=truth2[1];
truthU2:=truth2[2];
indeterminacyL2:=indeterminacy2[1];
indeterminacyU2:=indeterminacy2[2];
falsityL2:=falsity2[1];
```

```
falsityU2:=falsity2[2];
truthL:=max(truthL1,truthL2);
truthU:=max(truthU1,truthU2);
indeterminacyL:=max(indeterminacyL1,indeterminacyL2);
indeterminacyU:=max(indeterminacyU1,indeterminacyU2);
falsityL:=max(falsityL1,falsityL2);
falsityU:=max(falsityU1,falsityU2);
unionMat(i,j):=convert([[truthL,truthU],[indeterminacyL,indeterminacyU],[falsityL,falsityU]],string)
;
end do;
end do;
unionMat;
else
print("dimension of given matrices must be equal!");
end if;
end proc:
```

Example 3. Say that:

$$E=\begin{bmatrix} "[[1, 1], [.7, .8], [.1, .6]]" & "[[.2, .4], [.2, .8], [.1, .9]]" \\ "[[.8, .9], [.3, .5], [.1, .2]]" & "[[.1, .2], [.5, .7], [.2, .5]]" \end{bmatrix}$$

$$F=\begin{bmatrix} "[[0.7, 1], [.7, .8], [.4, .6]]" & "[[.2, .3], [.2, .6], [.1, .3]]" \\ "[[.2, .4], [.3, .5], [.1, .3]]" & "[[.1, .2], [.5, .7], [.3, .5]]" \end{bmatrix}$$

So, the union of previous matrices is done by calling the function:

Union(E, F);

And the result is:

$$E_{IVNM} \cup F_{IVNM} = \begin{bmatrix} "[[1, 1], [.7, .8], [.4, .6]]" & "[[.2, .4], [.2, .8], [.1, .9]]" \\ "[[.8, .9], [.3, .5], [.1, .3]]" & "[[.1, .2], [.5, .7], [.3, .5]]" \end{bmatrix}$$

3.6. Computing intersection of two interval valued neutrosophic matrices

The intersection of two interval valued neutrosophic matrices A and B is defined as follow:

$A \cap B = C = \left[< d_{ij_T}, d_{ij_I}, d_{ij_F} >\right]_{m\times\ n}$

Where:

$c_{ij_T} = \left[a_{ij_{TL}} \wedge b_{ij_{TL}}, a_{ij_{TU}} \wedge b_{ij_{TU}}\ \right],$

$c_{ij_I} = \left[a_{ij_{IL}} \vee b_{ij_{IL}}, a_{ij_{IU}} \vee b_{ij_{IU}}\ \right],$

$c_{ij_F} = \left[a_{ij_{FL}} \vee b_{ij_{FL}}, a_{ij_{FU}} \vee b_{ij_{FU}}\ \right]$

And this is done calling the function Intersection(A,B) is defined in the following manner.

```
Intersection:=proc(mat1::Matrix,mat2::Matrix)
m1,n1:=LinearAlgebra[Dimension](mat1);
m2,n2:=LinearAlgebra[Dimension](mat2);
if (n1=n2) and (m1=m2) then
m:=m1;n:=n1;
intersectMat:=Matrix(m,n);
for i from 1 to m by 1 do
for j from 1 to n by 1 do
x:=parse(mat1(i,j));
y:=parse(mat2(i,j));
truth1:=x[1];
indeterminacy1:=x[2];
falsity1:=x[3];
truthL1:=truth1[1];
truthU1:=truth1[2];
indeterminacyL1:=indeterminacy1[1];
indeterminacyU1:=indeterminacy1[2];
falsityL1:=falsity1[1];
falsityU1:=falsity1[2];
truth2:=y[1];
```

```
indeterminacy2:=y[2];
falsity2:=y[3];
truthL2:=truth2[1];
truthU2:=truth2[2];
indeterminacyL2:=indeterminacy2[1];
indeterminacyU2:=indeterminacy2[2];
falsityL2:=falsity2[1];
falsityU2:=falsity2[2];
truthL:=min(truthL1,truthL2);
truthU:=min(truthU1,truthU2);
indeterminacyL:=max(indeterminacyL1,indeterminacyL2);
indeterminacyU:=max(indeterminacyU1,indeterminacyU2);
falsityL:=max(falsityL1,falsityL2);
falsityU:=max(falsityU1,falsityU2);
intersectMat(i,j):=convert([[truthL,truthU],[indeterminacyL,indeterminacyU],[falsityL,falsityU]],stri
ng);
end do;
end do;
intersectMat;
else
print("dimension of given matrices must be equal!");
end if;
end proc:
```

Example 4. Find intersection of interval valued neutrosophic matrices E and F presented in example 3.

Solution:

Calling the function Intersection (E, F); yields to the solution:

$$\begin{bmatrix} \text{"[[.7, 1], [.7, .8], [.4, .6]]"} & \text{"[[.2, .3], [.2, .8], [.1, .9]]"} \\ \text{"[[.2, .4], [.3, .5], [.1, .3]]"} & \text{"[[.1, .2], [.5, .7], [.3, .5]]"} \end{bmatrix}$$

3.7. Addition of two interval valued neutrosophic matrices.

The Addition of two interval valued neutrosophic matrices A and B is defined as follow:

$A \oplus B = S = [< s_{ij_T}, s_{ij_I}, s_{ij_F} >]_{m\times\ n}$

Where:

$s_{ij_T} = \left[a_{ij_{TL}} + b_{ij_{TL}} - a_{ij_{TL}}.b_{ij_{TL}}, a_{ij_{TU}} + b_{ij_{TU}} - a_{ij_{TU}}.b_{ij_{TU}}\right],$

$s_{ij_I} = \left[a_{ij_{IL}}.b_{ij_{IL}}, a_{ij_{IU}}.b_{ij_{IU}}\right],$

$s_{ij_F} = \left[a_{ij_{FL}}.b_{ij_{FL}}, a_{ij_{FU}}.b_{ij_{FU}}\right],$

And can be done calling the function **Addition (A, B)** which is defined as follow:

```
Addition:=proc(mat1::Matrix,mat2::Matrix)
m1,n1:=LinearAlgebra[Dimension](mat1);
m2,n2:=LinearAlgebra[Dimension](mat2);
if (n1=n2) and (m1=m2) then
m:=m1;n:=n1;
addMat:=Matrix(m,n);
for i from 1 to m by 1 do
for j from 1 to n by 1 do
x:=parse(mat1(i,j));
y:=parse(mat2(i,j));
truth1:=x[1];
indeterminacy1:=x[2];
falsity1:=x[3];
truthL1:=truth1[1];
truthU1:=truth1[2];
indeterminacyL1:=indeterminacy1[1];
indeterminacyU1:=indeterminacy1[2];
falsityL1:=falsity1[1];
falsityU1:=falsity1[2];
truth2:=y[1];
```

```
indeterminacy2:=y[2];
falsity2:=y[3];
truthL2:=truth2[1];
truthU2:=truth2[2];
indeterminacyL2:=indeterminacy2[1];
indeterminacyU2:=indeterminacy2[2];
falsityL2:=falsity2[1];
falsityU2:=falsity2[2];
truthL:=truthL1+truthL2-truthL1*truthL2;
truthU:=truthU1+truthU2-truthU1*truthU2;
indeterminacyL:=indeterminacyL1*indeterminacyL2;
indeterminacyU:=indeterminacyU1*indeterminacyU2;
falsityL:=falsityL1*falsityL2;
falsityU:=falsityU1*falsityU2;
addMat(i,j):=convert([[truthL,truthU],[indeterminacyL,indeterminacyU],[falsityL,falsityU]],string);
end do;
end do;
addMat;
else
print("dimension of given matrices must be equal!");
end if;
end proc:
```

Example 5. In this example we find the addition of two interval valued neutrosophic matrices E and F presented in example 3 calling the function:

Addition(E,F);

$$\begin{bmatrix} "[[1.0, 1], [.49, .64], [.4e-1, .36]]" & "[[.36, .58], [.4e-1, .48], [.1e-1, .27]]" \\ "[[.84, .94], [.9e-1, .25], [.1e-1, .6e-1]]" & "[[.19, .36], [.25, .49], [.6e-1, .25]]" \end{bmatrix}$$

3.8. Product of two interval valued neutrosophic matrices

The product of two interval valued neutrosophic matrices A and B is defined as follow:

$A \odot B = R = [< r_{ij_T}, r_{ij_I}, r_{ij_F} >]_{m \times n}$

where

$r_{ij_T} = \left[a_{ij_{TL}}.b_{ij_{TL}}, a_{ij_{TU}}.b_{ij_{TU}}\right],$

$r_{ij_I} = \left[a_{ij_{IL}} + b_{ij_{IL}} - a_{ij_{IL}}.b_{ij_{IL}}, a_{ij_{IU}} + b_{ij_{IU}} - a_{ij_{IU}}.b_{ij_{IU}}\right],$

$r_{ij_F} = \left[a_{ij_{FL}} + b_{ij_{FL}} - a_{ij_{FL}}.b_{ij_{FL}}, a_{ij_{FU}} + b_{ij_{FU}} - a_{ij_{FU}}.b_{ij_{FU}}\right]$

Which is simply done by the call of the function **Product (A, B)** defined as follow:

```
Prod:=proc(mat1::Matrix,mat2::Matrix)
m1,n1:=LinearAlgebra[Dimension](mat1);
m2,n2:=LinearAlgebra[Dimension](mat2);
if (n1=n2) and (m1=m2) then
m:=m1;n:=n1;
prodMat:=Matrix(m,n);
for i from 1 to m by 1 do
for j from 1 to n by 1 do
x:=parse(mat1(i,j));
y:=parse(mat2(i,j));
truth1:=x[1];
indeterminacy1:=x[2];
falsity1:=x[3];
truthL1:=truth1[1];
truthU1:=truth1[2];
indeterminacyL1:=indeterminacy1[1];
indeterminacyU1:=indeterminacy1[2];
falsityL1:=falsity1[1];
falsityU1:=falsity1[2];
truth2:=y[1];
```

```
indeterminacy2:=y[2];
falsity2:=y[3];
truthL2:=truth2[1];
truthU2:=truth2[2];
indeterminacyL2:=indeterminacy2[1];
indeterminacyU2:=indeterminacy2[2];
falsityL2:=falsity2[1];
falsityU2:=falsity2[2];
truthL:=truthL1*truthL2;
truthU:=truthU1*truthU2;
indeterminacyL:=indeterminacyL1+indeterminacyL2-indeterminacyL1*indeterminacyL2;
indeterminacyU:=indeterminacyU1+indeterminacyU2-indeterminacyU1*indeterminacyU2;
falsityL:=falsityL1+falsityL2-falsityL1*falsityL2;
falsityU:=falsityU1+falsityU2-falsityU1*falsityU2;
prodMat(i,j):=convert([[truthL,truthU],[indeterminacyL,indeterminacyU],[falsityL,falsityU]],string);
end do;
end do;
prodMat;
else
print("dimension of given matrices must be equal!");
end if;
end proc:
```

Example 6. In this example we evaluate the product of the two interval valued neutrosophic matrices E and F presented in example 3 by calling of the command:

Product(E, F);

$$\begin{bmatrix} \text{"[[.7, 1], [.91, .96], [.46, .84]]"} & \text{"[[.4e-1, .12], [.36, .92], [.19, .93]]"} \\ \text{"[[.16, .36], [.51, .75], [.19, .44]]"} & \text{"[[.1e-1, .4e-1], [.75, .91], [.44, .75]]"} \end{bmatrix}$$

3.9. Transpose of interval valued neutrosophic matrix

Transpose of interval valued neutrosophic matrix simply done by calling of the function **Transpose(A)** defined as follow:

```
Transpose:=proc(mat::Matrix)
m,n:=LinearAlgebra[Dimension](mat);
temp:=Matrix(n,m);
for i from 1 to n by 1 do
for j from 1 to m by 1 do
temp(i,j):=mat(j,i);
end do;
end do;
temp;
end proc:
```

Example 7. In this example we evaluate the transpose of the interval valued neutrosophic matrix E presented in example 3:

Transpose(E);

$$\begin{bmatrix} \text{"[[1, 1], [.7, .8], [.1, .6]]"} & \text{"[[.8, .9], [.3, .5], [.1, .2]]"} \\ \text{"[[.2, .4], [.2, .8], [.1, .9]]"} & \text{"[[.1, .2], [.5, .7], [.2, .5]]"} \end{bmatrix}$$

4. Conclusions

This paper proposed new Maple package to do operations on interval valued neutrosophic matrices including complement, transpose, union, intersection, addition, product, sum and product of interval valued neutrosophic matrices. This package is very useful in neutrosophic decision making operations and on neutrosophic events simulation. In future work we are looking forward to generalize this package to other neutrosophic sets like fermatean neutrosophic sets and refined neutrosophic sets.

Funding:
"This research received no external funding."
Acknowledgments:
Authors are very grateful to the chief editor and reviewers for their comments and suggestions which improves the work.
Conflicts of Interest: "The authors declare no conflict of interest."

Data Availability Statement; The authors declares that there is no Data Availability

References:

1. L. Zadeh, "Fuzzy Sets," Information and Control, vol. 8, no. 3, pp. 338-353, 1965.

2. K. T. Atanassov, "Intuitionistic fuzzy sets," Fuzzy Sets andSystems, vol. 20, no. 1, pp. 87–96, 1986.

3. F. Smarandache, Neutrosophy. Neutrosophic Probability, Set, and Logic, USA: Amer. Res. Press, 1998.

4. S. Alias and D. Mohamad, "A Review on Neutrosophic Set and Its Development," Menemui Matematik (Discovering Mathematics), vol. 39, no. 2, pp. 61-69, 2017.

5. F. Karaaslan, K. Hayat and C. Jana, "The Determinant and Adjoint of an Interval-Valued Neutrosophic Matrix," in Neutrosophic Operational Research, 2021.

6. M. B. Zeina and A. Hatip, " Neutrosophic Random Variables," Neutrosophic Sets and Systems, vol. 39, pp. 44-52, 2021.

7. M. B. Zeina, "Neutrosophic Event-Based Queueing Model," International Journal of Neutrosophic Science, vol. 6, no. 1, pp. 48-55, 2020.

8. M. B. Zeina, "Erlang Service Queueing Model with Neutrosophic Parameters," International Journal of Neutrosophic Science, vol. 6, no. 2, pp. 106-112, 2020.

9. M. Abobala and M. Ibrahim, "An Introduction to Refined Neutrosophic Number Theory," Neutrosophic Sets and Systems, vol. 45, pp. 40-53, 2021.

10. R. Sherwani, M. Naeem, M. Aslam, M. Raza, M. Abid and S. Abbas, " Neutrosophic Beta Distribution with Properties and Applications," Neutrosophic Sets and Systems, vol. 41, pp. 209-214, 2021.

11. S.Broumi, S.K.Prabha, "Fermatean Neutrosophic Matrices and Their Basic Operations, " Neutrosophic Sets and Systems, Vol. 58, pp.572-595,2023 DOI: 10.5281/zenodo.8404536

12. S. Broumi, S. krishna Prabha, and Vakkas Uluçay, "Interval-Valued Fermatean Neutrosophic Shortest Path Problem via Score Function", Neutrosophic syst. appl., vol. 11, pp. 1–10, Oct. 2023.

13. S. Broumi, S., Mohanaselvi, T. , Witczak, M. , Talea, A. , Bakali, and F. Smarandache, "Complex fermatean neutrosophic graph and application to decision making", Decis. Mak. Appl. Manag. Eng., vol. 6, no. 1, pp. 474–501, Apr. 2023.

14. D. Sasikala and B. Divya, "A Newfangled Interpretation on Fermatean Neutrosophic Dombi Fuzzy Graphs", Neutrosophic syst. appl., vol. 7, pp. 36–53, Jul. 2023.

15. A. Salama, H. A. El-Ghareeb, A. M. Manie and F. Smarandache, "Introduction to Develop Some Software Programs for Dealing with Neutrosophic Sets," Neutrosophic Sets and Systems, vol. 3, pp. 51-52, 2014.

16. M. Bakro, R. Al-Kamha and Q. Kanafani, "Neutrosophication Functions and their Implementation by MATLAB Program," Neutrosophic Sets and Systems, vol. 40, pp. 169-178, 2021.

17. S. Broumi, N. D., A. Bakali, M. Talea, F. Smarandache, M. Lathamaheswari and J. Kavikumar, "Implementation of Neutrosophic Function Memberships Using MATLAB Program," Neutrosophic Sets and Systems, vol. 27, pp. 44-52, 2019.

18. M. B. Zeina , O. Zeitouny , F. Masri , F. Kadoura and S. Broumi, "Operations on Single-Valued Trapezoidal Neutrosophic Numbers using (α,β,γ)-Cuts "Maple Package"," International Journal of Neutrosophic Science, vol. 15, no. 2, pp. 113-122, 2021.

19. S. Broumi, M. B. Zeina, M. Lathamaheswari, A. Bakali and M. Talea, "A Maple Code to Perform Operations on Single Valued Neutrosophic Matrices," Neutrosophic Sets and Systems, vol. 49, pp. 485-508, 2022.

20. H. Wang, F. Smarandache, Y. Zhang and R. Sunderraman, "Single Valued Neutrosophic Sets," Multispace and Multistructure, vol. 4, pp. 410-413, 2005.

21. H. Zhang, J. Wang, and X. Chen, "Interval Neutrosophic Sets and Their Application in Multicriteria Decision Making Problems," The Scientific World Journal, vol. 2014, pp. 1–15, 2014. doi: 10.1155/2014/645953.

22. F. Smarandache, "The Score, Accuracy, and Certainty Functions determine a Total Order on the Set of Neutrosophic Triplets (T, I, F)." Neutrosophic Sets and Systems 38, 1 (2020).

Received: July 7, 2023. Accepted: Nov 22, 2023